Identifying Landmarks and Relations in Grounded Route Descriptions of IITK

Mohit Gupta and Subham Modi
Instructor: Dr. Amitabh Mukherjee
Indian Institute of Technology, Kanpur, India
{mohitt, smodi, amit} @ iitk.ac.in

April 16, 2013

ABSTRACT

Most computer generated routes today are based on the shortest path. Also, they rarely include landmarks in their descriptions. The problem our project tries to address is to compute route instructions on the basis of least complexity (such as less number of decision points) and trying to include landmarks and other features like an intelligent agent (human being) would do. To achieve the same, we collected a corpus of several textual route descriptions in natural language, for a randomly generated source and destination in IITK. Each route description was parsed using the Stanford Parser [5] and its output was then filtered to generate minimal command-like chunks. A Path is computed from this parsed and filtered output and its correctness is then verified manually.

1. Introduction

While computer generated navigation instructions are gaining more and more popularity, most of these compute routes on the basis of shortest or fastest paths. They tend to hide the information about landmarks and other salient features of the path such as landmarks and road geometry. However, a person's first level of spatial awareness develops on the basis of such features in the environment.

How are route descriptions given by an intelligent agent more effective from that generated by a machine?
An Intelligent agent uses several landmarks and other points of interest in the description and choice of path is also different depending on the complexity of decision points.
Example (refer fig 1): if someone wants to know how to get from Hall-2 to the Computer Center, it is most likely that one would ask him to go straight until the SAC crossing and take a left from there. Enter the gate on your left opposite to the auditorium. The Computer Center is about 50m straight after entering this gate.

At each decision point, a human way finder would be interested in the cognizable features of the environment. Notice how 'auditorium' and 'crossing' serve this purpose in the above example. Also, a shorter path to the computer center would be through the academic area but it would include a much more complicated nest of decision points and hence, is avoided.

"It is easier to follow directions if they are explained through a series of landmarks instead of street names. Landmarks play a central role in human spatial cognition. They are fundamental to the way humans learn an environment and construct mental representations of it. Because of their dominance in human mental representations of space, landmarks are widely used in human way finding and human communication about routes." (Including Landmarks in Routing Instructions, 2010)

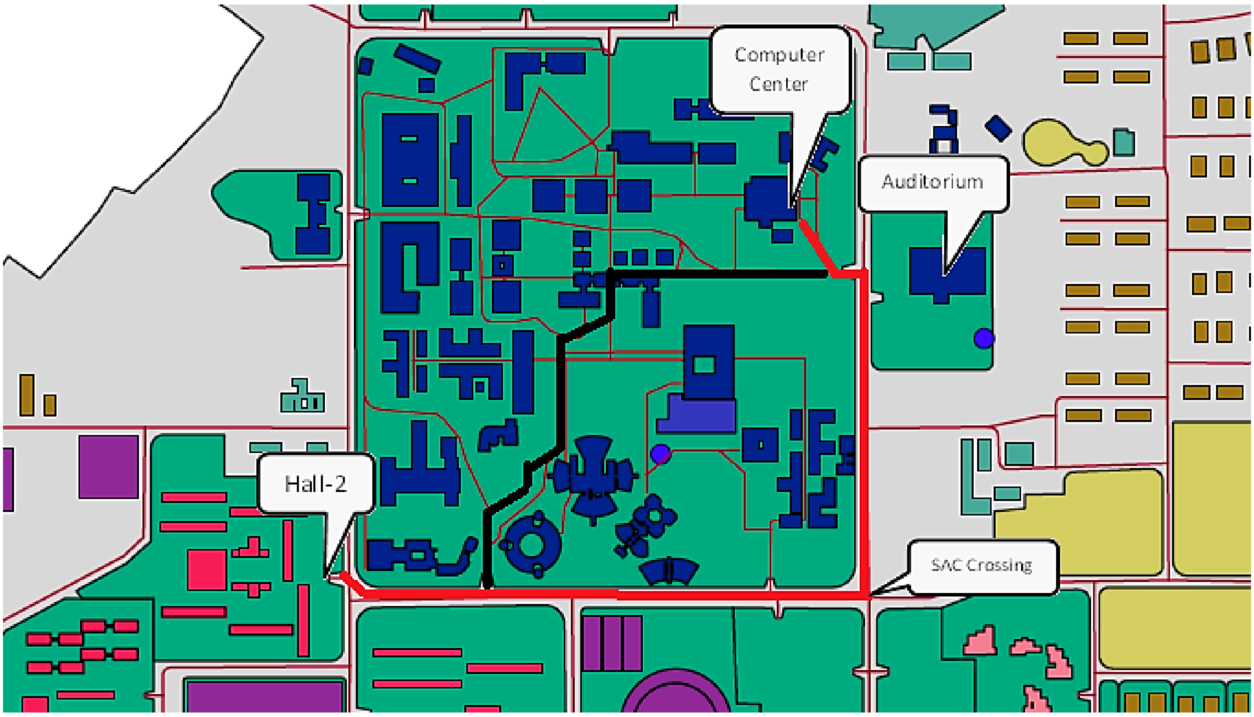


Fig. 1. Map showing the route from Hall-2 to the computer center

2. Relevant Work Done

"Matt Duckham and Stephan Winter from the Department of Geomatics at the University of Melbourne developed for Sensis for whereis.com, (An Australian web mapping and routing service owned by Sensis) a model for incorporating landmarks into routing instructions. The model relies solely on information about the types of landmarks present in the environment, in addition to the road network and route geometry. " [4]. The paper corresponding to their research is titled as "*Including Landmarks in Routing Instructions*" [1] which mainly focuses on developing a model for incorporating landmarks into routing instructions.

Varunesh Mishra, Sushobhan Nayak and Professor Amitabha Mukerjee from the Department of Computer Science and Engineering at IIT Kanpur, investigated three path finding heuristics for way finding in hierarchical maps. They looked into their relative efficacy in predicting human behavior by comparing their results against paths obtained from subjects for a campus map. They also tried to provide an explanation based on working memory hypothesis and the properties of the regionalized environment like the college campus to explain this trend. The paper corresponding to their research is titled as "*Towards a Cognitive Model for Human Wayfinding Behavior in Regionalized Environments*"[2] and its main emphasis is on developing a model for incorporating landmarks into routing instructions.

3. Tools and Data used, modified and created

Corpus set was created (with the help of different users) comprising of several inputs, between a randomly generated source and destination, in the form of Natural Language. In addition to this, *Stanford PCFG Parser* was used from the source [3] to parse the generated input into a dependency tree.

Map of IITK (.shp format) from [5] was used and then modified to include only major landmarks and roads. The road map was further altered to maintain atomicity between two junctions. The start and end coordinates of each atomic road were lifted from the attribute table and compiled separately into a database.

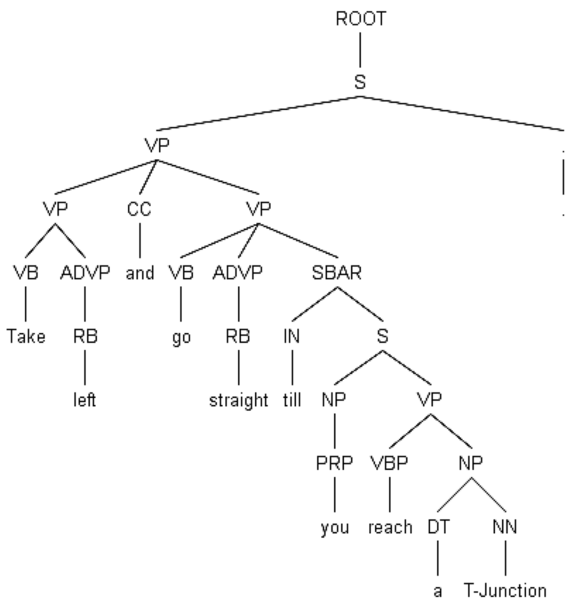
4. Algorithm

To make a machine understand and be able to trace a route description on the map, it is important that the natural language text be fragmented and then mapped to a finite set of minimal instructions. We achieved this by using a two level approach.

In the first level, we used the Stanford Parser, which fragments a sentence into several phases (noun phase, verb phase etc.) and tags each word as a preposition, proper noun, verb etc. and outputs as a dependency tree as shown in Fig. 2(left).

In the second level, we tried to sieve out a minimal set of words (refer Fig. 2(top)) from each sentence, such that all important information like spatial markers such as left, right adjacent to etc. is retained. Locating landmarks is easier, since, they are almost always the proper nouns in the sentences.

The filters were improvised using the feedback of their application over the corpus entries.

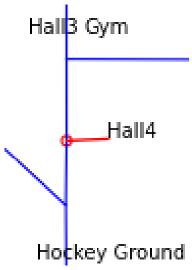



```
turn left
--------------------
go straight
till t-junction
--------------------
```

Fig 2. Stanford Parser Dependency tree (left)
Filtered minimal output (top)

Table 1. The set of identifiers our sieved and minimalized outputs contain is by and large:

Turn left/right	-	Take a left/right from here
go straight	-	Keep moving in the same direction from here
go north/south/east/west	-	Move in the north/south/east/west direction
from <landmark/spatial feature>	-	<landmark/spatial feature> is located here
till <landmark/spatial feature>	-	Move till <landmark/spatial> feature is reached
follow path	-	Trace path until a decision point is reached
passing <landmark/spatial feature>	-	Path would contain < landmark/spatial feature>
For <distance/time>	-	Move for <distance/time>



x-start	y-start	x-end	y-end	split id
423346.1	2932214	423346.1	2932180	52
423346.1	2932180	423346.2	2932070	53
422995.3	2932003	422995.5	2931966	54
423300.6	2931878	423346	2931878	55
423801.4	2932953	423799.8	2932737	57
423181.1	2932362	423178.9	2932310	58
422992.6	2932241	422995.6	2932153	60

Landmark	x-gate	y-gate	x-mid	y-mid
SAC	423721.1	2932449	423723	2932391
New SAC	423300.6	2931878	423296	2931841
GH1	423569.3	2932180	423570.4	2932135
Hall1	423346.1	2932345	423397	2932346
Hall2	423346.1	2932459	423279.5	2932467
Hall3	423346.1	2932214	423289.4	2932214
Hall4	423346.2	2932070	423412.3	2932073

Fig. 3. (a) landmarks (b) snippet of roads database (c) snippet of landmarks database

Now, we generated a database with roads as entries (see Fig 3(b)), storing the two end 2-D coordinates (in m) of each road and a 'split id' (generated while atomizing the road network by splitting roads at all intersections and turns). For simplicity, we ignored some areas of the map and tried to keep the most significant ones. For the landmark's database (see Fig. 3(c)), we assigned their locations as the location of their entry. This coordinate would be on the road network itself and hence, would not give us any information about the relative location of the landmark. So, we assigned another coordinate away from the gate in the direction of the landmark's geometric center (refer Fig 3(a)). For the sake of simplicity, we avoid using the four corners of the landmark's bounding box, though it would be a more realistic approach in some measure. We also maintained a synonym's database for the landmarks. Any search would first go through the synonym's database and if found at some index, look for the same index in the landmark's database to get the location.

We know the source and the destination corresponding to the route description in the any entry in our corpus. Hence, we assign our starting location to the coordinate of the gate of the source, which we can be found from the landmark's database. Let us call the current coordinate as 'here'. We next search for the road(s) which can be accessed from 'here'. We now check that the assigned task (move in some direction or pass some landmark or reach a T-junction etc.) is satisfied by moving to which of the accessible locations from 'here'. Just after the first step, what we have is only our starting location. Left/right are not defined, since, we are unaware how the traveler's face is oriented. In order to overcome the ambiguity, we assumed that the traveler would be facing opposite to the source landmark.

We assume any two points at less than 3 meters to be the same (reasonable since the shortest road on the map is 35m long). We define a landmark to be nearby if its location lies within 30m from 'here'.

```
C:\Python27\python.exe
Source: nursery
Dest: vh
Move Towards Landmark or Direction(e.g. N, NW, L/R etc): go straight till vh
Near Audi
Acad Area Gate(CC)
Audi
Near Acad Area Gate(CC)
SAC Circle
Outreach
Media Lab
Move Towards Landmark or Direction(e.g. N, NW, L/R etc): _
```

Fig. 4. Sample Program run

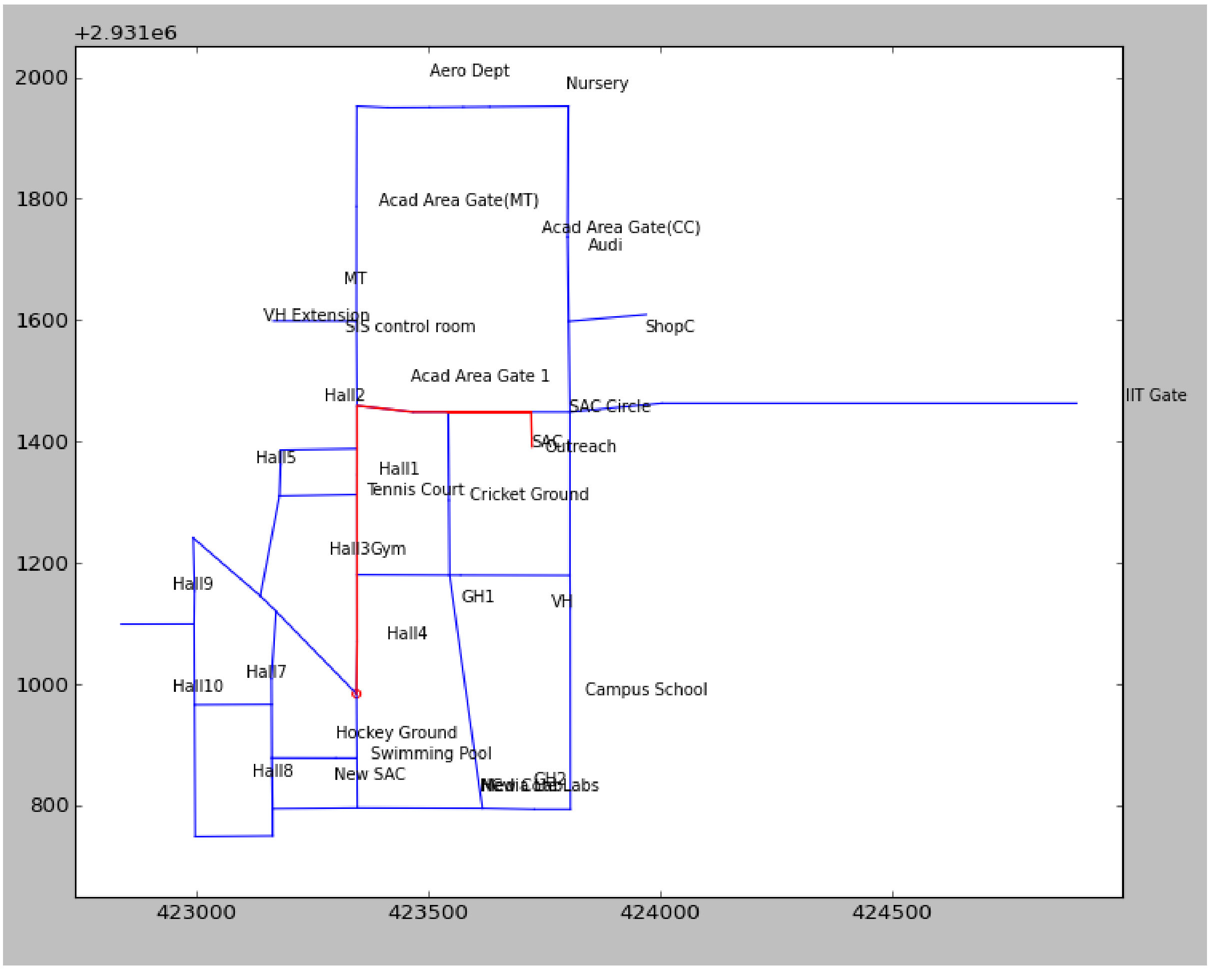


Fig. 5. Map highlighting the route (red track) and the current location (red marker)

5. Results and Conclusions

We have been able to feature the understanding of almost all the identifiers in our program. The code also shows the IITK map, highlighting the route and the current location, which are updated in real time (see Fig, 4 and 5). It also shows the landmark(s) accessible from 'here', landmark(s) nearby and the landmark(s) we crossed when we moved from the previous location to 'here'. We have also included the facility to add a landmark at any location. Adding a synonym to the synonym's database, given one of the names of the landmark has is also been implemented.

The parsing and filtering script performs very well over our corpus entries. It works correctly on the current 68 route descriptions. We also verified the route computed by the scripts combined to be correct for almost all the 68 entries. Hence, we achieve our goal of training the computer to understand route descriptions in natural language given by an intelligent agent. (We have successfully made the computer intelligent in cartography!)

6. Future Work

Apart from the features that were implemented in our project, one can learn that how more popular or more frequently visited landmarks and roads in the route descriptions corpus can play a major role in the description of a route between a given source and destination. We could generate simpler and more cognizable routes by assigning a lower cost more popular roads and landmarks, blending this cost with the distance cost and using heuristics like less number of decision points. The computed route could also be output in the form of natural language text. Such a vivid textual description, we believe, would be much more efficient because of eliminating the necessity of carrying a map to find an address.

We worked using only the route description comprising of only those landmarks that were present in our map and relevant databases, but on a future note, one can also consider and include new landmarks from route descriptions that weren't demarcated earlier in the map by guessing the location of an unknown landmark that is come across in a route description, deciding whether it is a new landmark or a synonym of an existing one and hence, adding it to the database. This would lead to a self-sustaining map database.

7. Acknowledgements

We thank our instructor Professor Amitabha Mukherjee for advising us throughout our project and motivating us towards this problem. We thank Professor Bharat Lohani for providing us with the electronic map of IITK. We would also like to thank Arbaaz Khan for providing needful assistance during the initiation of the project.

8. References

[1] Winter, S.; Duckham, M. and Robinson M. 2010, Including Landmarks in Routing Instructions, Journal of Location Based Services 4 (1), pp. – 28-52

[2] Nayak, S.; Mishra, V.; and Mukerjee A. 2011, Towards a Cognitive Model for Human Wayfinding Behavior in Regionalized Environments, AAAI Fall Symposium – Technical Report FS-11-01, pp. 249-256

[3] http://nlp.stanford.edu/software/stanford-dependencies.shtml

[4] www.geom.unimelb.edu.au

[5] www.geokno.com/products/largemap.php

____*All codes used in the project were our own and not taken from anywhere.*