Dialog-Based Location Unaware Wayfinding

A dissertation submitted
in Partial Fulfillment of the Requirements
for the Degree of

Master of Technology

by

Arbaz Khan

to the
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
May 7, 2014

CERTIFICATE

It is certified that the work contained in the dissertation titled **Dialog-Based Location Unaware Wayfinding**, by **Arbaz Khan**, has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Prof Harish Karnick
Computer Science and Engineering
IIT Kanpur

Prof Bharat Lohani
Department of Geoinformatics
IIT Kanpur

Prof. Stephan Winter
Department of Geomatics
University of Melbourne

May 7, 2014

ABSTRACT

Name of student: **Arbaz Khan** Roll no: **Y9227128**

Degree for which submitted: **Master of Technology**

Department: **Computer Science and Engineering**

Thesis title: **Dialog-Based Location Unaware Wayfinding**

Names of Thesis Supervisors

1. **Prof Harish Karnick**
2. **Prof Bharat Lohani**
3. **Prof. Stephan Winter**

Month and year of thesis submission: **May 7, 2014**

I plan to finalize the text of the abstract in the end of the writing.

Contents

Chapter 1

Introduction

Chapter 2

Problem Definition and Related Work

2.1 Dialog Based Wayfinding Systems

Significant research has been done in the area of defining [21] and generating [4, 8, 23, 27] good quality route instructions which are specialized for human understanding. These models have evolved from those working on handcrafted data disseminating turn-by-turn instructions in a one-step-per-sentence mappings to the systems processing data from Geographic Information Systems (GIS) using visible features of the surroundings and generating compact and more human-like descriptions. Yet, most of these systems were monologic systems in which the instructions are provided in advance or even incrementally, causing them to suffer from poor comprehensions and an inability to resolve user's ambiguities along the path. Owing to the benefits of a dialog system to respond to user queries and possibility to react to inadequate or poorly understood instructions, research works [10, 13, 29] had moved towards building dialog-driven wayfinding models. The major concern in these systems is the balance between not overwhelming the user with too much media to minimize the cognitive overload, and to allow several levels of details to produce comprehensive and accurate set of descriptions preventing any ambiguities or confusions. Although some [29] of these systems rely on purely speech-based interactions while others [10, 13] have used the luxury of realizing a multimodal interface which offers the user a freedom of choice by combining speech, pen and graphics based input signals. Although multimodal systems do help in matching user preferences and result in better and disambiguated input signals, but these systems lose the fundamental requirement of wayfinding assistance, which is, to discourage distractions from driving task and reduce the driver workload. Pure-auditory based route instructions help in servicing systems which cause to minimize the interference to driving task and thus create no negative impact on reaction times [31].

At present, there are only a handful of systems specializing in spoken dialogue-systems for wayfinding. These systems can be classified into three categories depending on the target domain - indoor wayfinding, human-machine interaction and outdoor wayfinding. The systems [5] based upon human-machine interaction aim to facilitate the machine in execution of spatial task by providing route information and handling questions raised by it on detecting ambiguities. These tasks though certainly are similar to real world wayfinding assistance models in dialog structure and end goals but differ in the ability of their subjects. Humans follow subjective and non-uniform speed-patterns, instruction understanding behavior and conceptual memory models.

For outdoor wayfinding, Janarthanam et. al. [12] introduced a mobile dialogue application that combines GIS systems with spoken-dialog interface to assist tourists in city navigation. It allows tourists to initiate tasks such as entity search(e.g. a museum or a restaurant), navigation assistance, entity inquiry and others. The proposed architecture is capable to work in a fairly complex spoken-dialog interaction (e.g., Take me to Hume) by the help of its semantic parser and modules for natural language generation (NLG) and understanding (NLU). Richter et. al. [29] worked on generating context-based route instructions for outdoor wayfinding. Using a spoken dialog interface, the proposed conceptual model allows user to tune the details of the route instructions from incremental turn-by-turn instructions to a more abstract set of instructions focussing upon destination descriptions. The model identifies and adapts to wayfinder's spatial knowledge about street network after deciding upon the prominence of a spatial entity derived from a hierarchy of visual, structural and cognitive features. Spoken-dialog systems for indoor wayfinding systems have not been so far realized as per our knowledge. Cuayáhuitl et. al. [3] worked on a text-based dialogue system for indoor wayfinding which can respond to user queries on locations in a building. The authors claimed that a dialogue system for indoor wayfinding with only text-based interaction leads to a very high overall scores of user satisfaction. Despite the fact that indoor settings are perfect for application of location-unawareness, yet none of these systems have yet attempted it and rely upon certain location sensitivity (such as support for infrared and wireless media) of the end-device. Also, the relatively less emphasis on evolution of pure spoken-dialog systems can be understood owing to their limitations in poor speech recognition and unfulfilled linguistic expectations.

In this work we try to overcome these limitations by setting up a cross-lingual communication platform to convey non-ambiguous route instructions. Furthermore, questions posted to the user for location tracking are such that the responses are recognisable speech commands (such as 'yes','no') and require primitive speech processing.

2.2 Evaluation Techniques

Simulations in a virtual environment provide an important means for evaluation and improvement of real-world systems. Once the virtual environment is set-up, the next task is to identify the metrics for evaluating effectivenss of the system. The work done in this thesis is closely related to natural language generation (NLG), only that we focus only on the spatiotemporal content of the linguistics. Thus, the technique and metrics used for evaluating NLG systems are relevant in this context. For example, if the number of users who reach the target are less in number, it reflects poor quality of NLG involved and thus could be related to the underlying spatio-temporal content of the instructions. Its only recently that increasing number of efforts have been made for evaluation of NLG systems. As Spanger et. al. [30] point out, there can be two categories of methods to evaluate the effectiveness of a NLG systems - intrinsic and extrinsic. *Intrinsic* methods compare the system with a benchmark/gold standard system, and *extrinsic* methods which use metrics to identify the goodness of the system such as the number of mistakes made by the user while following the instructions [34].

Further, research on evaluation of NLG tasks and investigation of issues moved towards introducing shared task challenges [6]. In this task, the evaluation is assessed with the help of performance of human users who follow the instructions of the NLG in a virtual environment with several rooms and corridors. The performance evaluation uses a set of objective as well as subjective measures. Objective measures include the distance travelled and time taken in reaching the goal as well as the number of instructions provided and the number of words per instruction. While, subjective measures provide a subjective rating of these instructions whether the instructions were useful. An interesting area of research [11] then emerged which works on modelling navigation assistance with real world problems like uncertainty in location and noisy feedback signals from the user. These works exploit the utility of virtual environment to eliminate the time and costs vested in real world experiments as well as provide a means to manipulate the environment targeting specific issues and context.

In this work, we move beyond the above mentioned virtual environments tested on humans. With the virtualization of spatial environment, we also virtualize the user via driver modelling and use a set of goodness metrics to measure the effectiveness of the system. Independent works have been done in the area of simulating driver behavior to study traffic and accident patterns but none of these have been integrated into a testbed for wayfinding assistance.

Chapter 3

Conceptual Model For Wayfinding

We introduce our location-unaware wayfinding model through a generic spoken-dialog based model for wayfinding. This model can be transformed to a location-unaware wayfinding model by specializing the module structure. The aim of this model is to assist wayfinding tasks using natural language dialogues. The next section presents an architecture of the model. We then propose a natural language independent communication protocol for conveying route-instructions in wayfinding assistance.

3.1 Architecture

The generic wayfinding model is designed to incorporate any linguistic structure and a predefined technique for user localization. It should also be personalized for user and should be able to identify and react to the context realizing the differences and constraints. The knowledge base behind the functioning should be dynamically updateable to incorporate the changes in spatial environment (such as broken road links, new buildings, etc). A pipelined architecture realizing the requirements is shown in Figure 3.1. First, a *user* (real human or modelled) requests for assistance to his destination. The *Natural Language Module* parses the semantics of utterances, processes spatial information and records a new query into the system. Before route instructions could be generated, it is required to locate the user (obtain the source

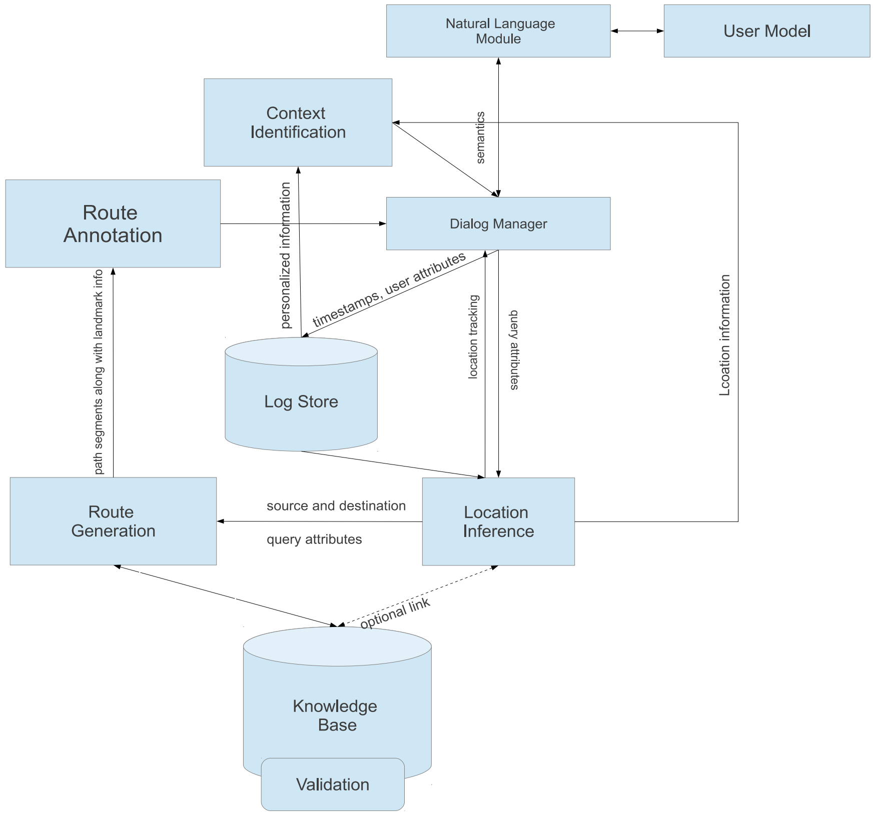

Figure 3.1: Architecture for a generic wayfinding model

location). Most of the wayfinding systems today use GPS systems for this purpose in an outdoor environment and a wireless or infrared media in indoor settings. Baus et. al. [1] introduced an adaptive model for localization that alternates between the different sensing technologies to overcome their individual limitations. For the system introduced in this work, we seek the user's location via dialog-based tracking, posing simple questions using features of a spatial environment. Nonetheless, we can abstract a module specifically for *location inference* which either can work independently (by using sensors or receptors) or uses the *knowledge base* of the system for localization. Thus, the knowledge base stores a characterization of the complete spatial environment. It also stores transportation network for target modalities (e.g., road links for vehicles, abstract spaces for pedestrians) and its relationship with the environment features (e.g., landmark-street association).

Once the information required to compute the topological route is collected, then the desired route is computed using query attributes (such as preferred route option like shortest route, scenic route, etc. and modality constraints like roads permissible for 4-wheelers). This computed route is passed along with the nearby landmark information for *route annotation*. The route annotator processes the salience of the landmarks, removes redundant spatial relationships and fills up the spatial content of the route instructions. Besides operating on the processed information, the system is also sensitive to the context i.e. user attributes (such as modality, speed patterns) and location information. The context needs to be considered before generating natural language route instructions. With the help of contextual information, a decision is made by the *dialog manager* on the delivery of route instructions to match with the time when a user needs to know it. So, decisions like these make sure that turn instructions are temporally matched with user's movement. The dialog manager also compiles the route instructions into a form understandable by the natural language module, which then serves to eventually fulfill the user request.

3.2 Communication Protocol

In this section we introduce a communication protocol to facilitate in a cross-lingual platform development. The protocol is targeted to deal with two kinds of scenarios - simple intersections and complex intersections. These latter scenario relates with the challenge of dealing intersections with larger number of possible actions (more than 4) and tackling ineffectiveness of standard direction models. but also on In a similar work, Klippel [15] introduced a formal representation of turn directions at decision points which could be chunked to produce better quality route instructions and could be tailored as per user preferences. But the formal theory model targets only simple intersections. Our approach is based upon similar notational representation of turn instructions but extends to complex intersections as well. The goals in designing this protocol are:

1. **Completeness** The protocol should indicate the action to be taken at each decision point implicitly or explicitly. A set of good quality instructions avoids the need of specifying action at each decision point by chunking action be-

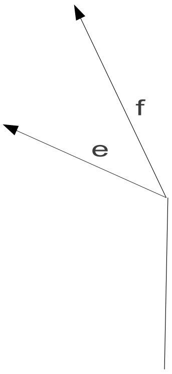

Figure 3.2: Ambiguity arises to determine the intended action when the route instructions are based on standard direction models as *take a left*.

haviour. For example, instructions '*go straight*' and '*take a left turn*' can be compactly presented as '*take the second left turn*' which implicitly directs action behaviour at each decision point and is arguably more comprehensible.

2. **Non-ambiguity** An ambiguous language model leads to confusion in action required at each decision point and thus could lead to disorientation. For instance, instructing to take a left turn in a topology as shown in Figure 3.2 is ambiguous and its not sure whether to take a left at e or f.

3. **Applicability** Alongwith completeness and non-ambiguity, its equally important to consider the ease of applying the communication protocol in real-world scenario. This means that the protocol should be structured such that it could be translated easily to the desired natural language.

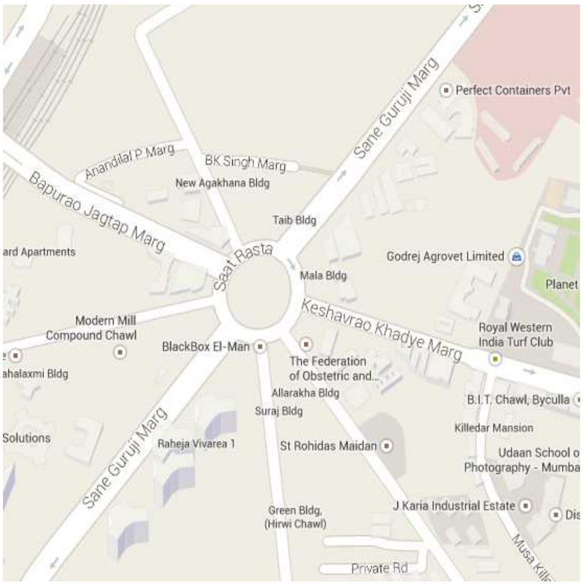

Figure 3.3: Real world example of a complex intersection in Mumbai, India taken from Google Maps [26].

The communication protocol proposed is an adaptive approach to formal representation of route instructions. It uses standard direction model like that of Klippel [15] when the resulting indications are non-ambiguous. However, if the turns are complex, it uses a clock-based convention to represent turn directions. The protocol uses a language independent symbolic encoding for landmarks and turn instructions at decision points. The two scenarios of simple and complex intersections differ in the symbolic encoding of directions while landmarks use a notational representation and each of these is elaborated below.

3.2.1 Landmarks

The landmarks are represented by notational IDs and by the direction (left or right) in which user encounters this landmark while moving on a path segment. So, if X is the notational ID of a landmark and if moving on the directed path segment, the user would encounter X on his right, then the corresponding representation is X^R. Similarly, X^L indicates that moving on the path segment, user can see X on his left. These representations along with that of turn behaviour at intersections (simple and complex) form the communication protocol.

3.2.2 Simple Intersections

Most of the intersections in the real-world fall under this category and are fairly trivial to communicate. To formally define a *simple intersection*, we divide field-view of the navigator in four triangular zones (see Figure 3.4). Choosing reference axis as the left direction perpendicular to the incoming road segment p, the left zone spans 45 degrees on either side of the reference point. Its mirror image in the field-view w.r.t. the decision point is the right zone. Similarly, one can conceptualize the other two zones. We define a decision point as *simple intersection*, if in each zone of the navigator's field view, there is atmost one road segment emerging from the decision point.

The instructions that are associated with change in direction are represented by

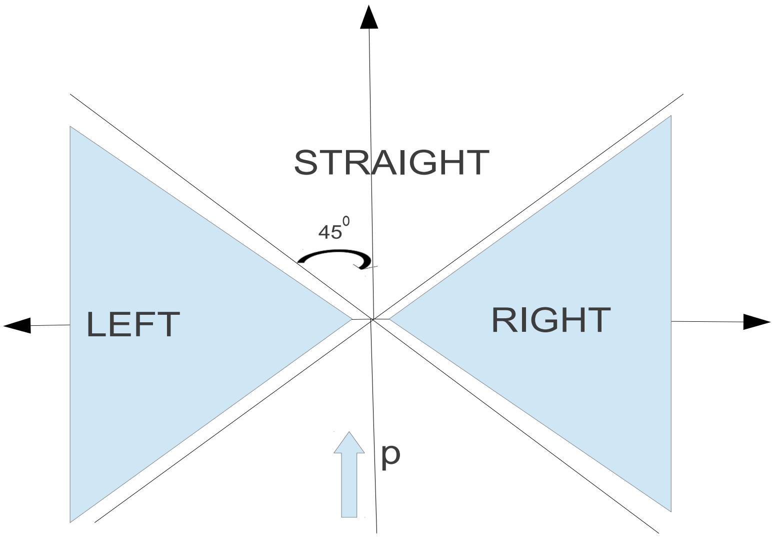

Figure 3.4: A decision point is simple intersection, if in each zone of the navigator's field view, there is atmost one road segment emerging from the decision point.

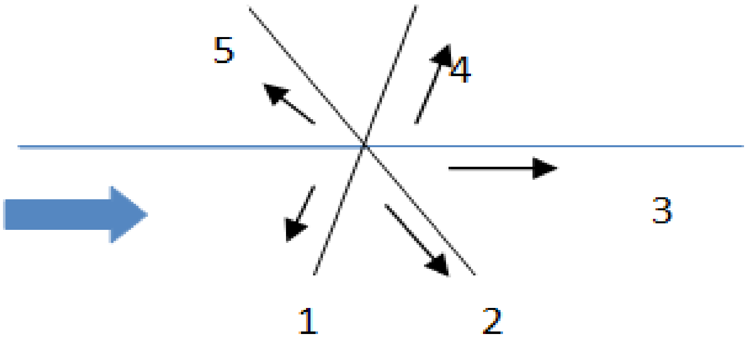

Figure 3.5: Clock based convention to non-ambigously represent directions at complex intersections

symbols **R** and **L**. The non-turning instruction indicating to go straight is represented by **S**.

3.2.3 Complex Intersections

Any decision point which is not a simple intersection is a *complex intersection*[1]. For representing complex intersections, we use a clock-based notation for a non-ambiguous conceptualization. Klippel [15] handles non-standard turns by using 8-sector model which opens up alternate notations such as **hl** (half left), **vr** (veer

[1] See Figure 3.3 and Figure 3.5

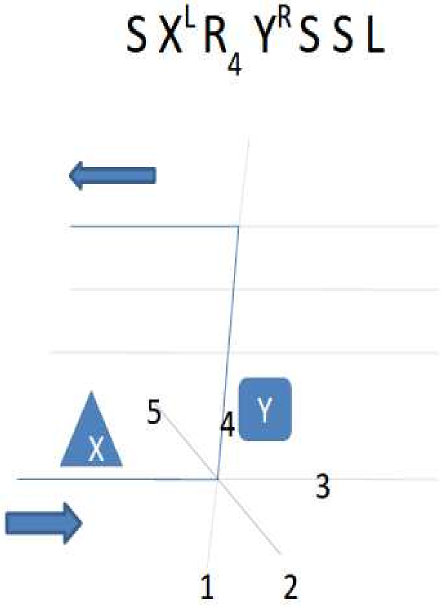

Figure 3.6: An Example to showcase an application of the proposed communication protocol. The intended route is represented symbolically in text at the top of the image. One possible translation of the encoding in english language can be as: 1. *Go Straight* 2. *You would find X on your left* 3. *At the intersection, take the 4th link in anti-clockwise direction from the most adjacent turn at your right side* 4. *You would find Y on your right* 5. *Take the third left after Y*

right), etc. These work well for 4-way intersections and 3-way intersections but as the number of emerging road segments increase beyond 4, the 8-sector model fails as there can be more than one road segment in the same sector.

A clock-based notation doesn't work on the basis of sector model and can handle extended multi-way intersections. The roads are represented in an anti-clockwise numbering starting from the first turn right-adjacent to incoming road segment as shown in Figure 3.5. Notationally, we represent complex turns in the form R_i, which represents the i^{th} numbered turn in anti-clockwise direction starting from the most adjacent turn on your right.

Figure 3.6 elaborates a full-fledged example combining the representations of landmarks and intersections and attempts to present an english translation of the same. Before we conclude, we claim that the qualitative calculi used by Klippel [15] on formal representations can be applied likewise to the proposed communication protocol as it is as a direct extension to the theory by including complex intersections. So if a route has no complex intersections, our communication protocol is semantically similar to that of Klippel. Continuing on the example from Figure 3.6, the route segment corresponding to the representation SSL is translated as *the third left* which is similar to the chunking rule of Klippel's *wayfinding choremes* [15], where by the help of term rewriting, an intermediate representation is deduced from simple representations, prior to natural-language translation.

Chapter 4

Dialog-Based Localization

In the previous chapter, we proposed a communication protocol to convey turn behaviour at every intersection. In this chapter, we introduce the algorithm for dialog-based tracking between the intersections to determine user's location and temporally align the delivery of route instructions with the user's movement and confirm his orientation. While posing questions to a user for localization and confirming orientation, we propose the use of an alternative reference to landmarks which highlights the distinctive geometry features. We next present a method to extrapolate movements of a user for location inference using speed predictions. To further cope up with errors and misinterpretations, we also introduce the algorithms to detect and resolve disorientations. The localization alogrithm along with the communication protocol are the two mutually exclusive and exhaustive components of the proposed location-unaware dialog-system.

4.1 Introduction

While guiding a user to his next path segment, its required to keep track of the user's location to avoid any disorientation. Since, there is no device-based support for location sensing, the only way to localize a user is by asking him about his location in a controlled input format. Though, researchers [16, 17, 24] have been working on processing unrestricted NL input to extract spatial information but none of these have been able to overcome the classical information extraction errors limiting the output. Further limitations brought by poor speech recognition add to the ineffectiveness of using unrestricted language as input. For this work, we try to limit the questions to those with an objective reply (such as yes or no) or recognisable speech commands (such as color of a building, etc.). These questions are strategically based on the salient features of the spatial environment i.e. *landmarks.* The other requirement to provide quality user interface is to be able to predict the correct position of the user with a good enough estimate, such that the number of questions asked to confirm the orientation and localize the user are kept to minimum. The 'number of questions asked',as we would discuss later in detail in Chapter 6, remains as a prime evaluation criteria.

4.2 Alternative Reference to Routemarks

In any session of wayfinding, a user is incrementally guided for turn behavior at every intersection. Between every two intersections, user is prompted for localization via questions based on whether he encountered the associated landmarks en-route. [1]. Lovelace et al. [21] distinguish between the landmarks according to the purpose they serve i.e. either in choosing the action at a decision point (landmarks) or confirming reorientation on a path segment (*routemarks*). In localization, landmarks are always treated as routemarks and with this, there is a difference in how they are communicated to the user. In the following discussion, the term landmark would be used substitutively for a routemark.

A landmark is referred to by its name (e.g., Eiffel tower) or by its category (e.g., hospital, T-junction) or both, depending on whichever defines its distinctiveness in its locality. Although, in general category-based landmarks are easily comprehensible under route instructions but similar familiariaty is not guaranteed with name-based landmarks unless the name is explicitly mentioned in a readable form (like for hotels). In such cases, it is preferable to refer a landmark by its distinctive

[1] The issue of how landmarks are associated to a path segment are discussed in Section **??**.

geometry feature (if any or else choose a landmark by geometry). For example, instead of refering to that building as *Visitor's Hostel*, the system should refer it alternatively as the "red colored 2-floored building" if its the only one so in the locality. Since, even a category-based landmark can be misinterpreted, esp. when its structure doesn't directly reflect its category (like a movie theatre might not look like a theatre), we reference a routemark by its distinctive geometry feature if its salience falls below a certain threshhold. The threshhold parameter differs for name-based landmarks based on the assumption that the chances of unfamiliarity to a name are more than the mismatch of landmark structure with its category.

4.3 Extrapolating User Movements

4.3.1 Estimating User Speed

Consider the situation when the user is on a particular path segment and is being guided to his destination through an incremental set of instructions. At this point, the instruction pertaining to the next intersection has been made known. The prompts are such that the user is required to give a positive response only when he sees a particular landmark after taking the needed action. For example, natural language equivalent (in English) to such a prompt could be, *Go straight at the next intersection and prompt me 'Yes' when you see a cafeteria on your left.* These instructions as discussed earlier are a part of the communication protocol. At this point, if the user follows the instruction correctly and takes a non-turning action at the next intersection, he should see the *cafeteria* on his left after some point of time. Since, the instructions do not explicitly or implicitly mention about any intersection in between, it is guaranteed that between the *cafeteria* and current location of the user, there is exactly one intersection. The time when a positive response is recorded from the user, the *cafeteria* becomes the new current location of the user. Thus based on distance measures and the corresponding time difference, speed can be computed. Thus, speed estimation can be put as:

$$User\ Speed = \frac{Distance\ between\ two\ recorded\ prompts}{Time\ difference\ between\ the\ two\ prompts}$$

4.3.2 Predicting User Speed

Speed predictions lead to adaptive nature of the localization algorithm and assist in extrapolating movement patterns of a user as per his speed profile. Continuing from the above example, consider a case when user doesn't respond with a 'Yes' prompt for a relatively long time. If the system acts passively waiting for the 'Yes' prompt, a disoriented user may be totally lost (unless the system accepts unrestricted NL to process a general user query). Thus, there needs to be a predefined time limit in which the user is expected to provide a response. If this time limit expires, the sytem can interrupt to generate another prompt confirming his orientation. Thus, to get an estimate of this time limit, it is required to predict speed of the user.

The speed predictions are done specific to a road segment and time of day. Some roads facilitate faster speeds than the others but the differences differ over the time of day. Also to be considered is the speed profile of the user. A slow driver might prefer to drive slow proportionately in every speed-limit scenario. Summing up, the algorithm for speed prediction works over two predictors -

- historical average speed on the road segment at this time of day ($H_{e,t}$),
- average relative speed deviation of the user from the historical speeds (A_u),
- dynamic slowdown factor (f).

For every upcoming road segment, speed is predicted using historical average speeds on the roads and the average deviations of the estimated speed of the user in this session from the historical average speeds on the travelled roads. The feature historical average speeds is stored specific to discretized time slots in a day owing to the observation that a road might be distinctly fast in the late nights as compared to that in peak-hours. The average relative speed deviation feature is related to the characteristic of the user and can help identify driving preferences. This gives a slow driver enough time gap between successive prompts for confirming orientation, while a fast driver might get different prompts to spatio-temporally match his rapid movements.

The third predictor is meant to consider live traffic information. There may be cases when a road has history of fast vehicle operating speeds and yet, temporary slowdowns maybe observed even for drivers with high-speed driving preferences. Such slowdowns usually occur due to natural interruptions such as foggy weather, torrential rains or snowfall, or even man-made interferences like festival celebrations, rallies or occassional traffic-jams due to miscellaneous reasons. The estimation of dynamic slowdown factor is done freshly at each time slot by averaging relative slowdowns at the given edge from all the users and can be mathematically modelled as:

$$f_e = \frac{1}{|U|} \sum_{u \, \in \, U} (1 - \frac{(A_u - A_{e,u})}{A_u})$$

$A_{e,u}$ – speed deviation of user u *at* e from historical average speeds at e at given time slot

A_u – average relative speed deviation of the user u from the historical average speeds

U – the set of users who travelled the edge e in this time slot

f_e – slowdown factor for edge e at given time slot

4.4 Detecting Disorientation

As discussed before, exactly one prompt is posed to the user between every two intersection points. This prompt asks the user to confirm his position and is put as early as possible at a road segment to detect disorientation early. There are two cases possible if no response is received from the user within the time limit (set after speed prediction). These cases are handled as below. Algorithm 1 presents the algorithm to detect disorientation.

Algorithm 1: DetectDisorientation(response,landmark,timeFrame)

```
input  : Response received at the end of time limit (can be None),
         the landmark position expected and
         current time limit
output: 1, if detected disorientation, 0 otherwise

if response is positive then
    go to next route segment
    current location becomes this landmark
else
    prompt to ask if the intermediate intersection was crossed
    $response \leftarrow getInput()$
    if response is positive then
        $timeFrame \leftarrow timeFrame \times waitFactor$
        // wait for timeFrame
        $response \leftarrow getInput(timeFrame)$ // blocks for timeFrame units
        if response received and response is positive then
            return 0
        else
            // Here when no response received or negative response
            return 1
    else
        return 0
```

Case-I - Slow Driver

If the user is unable to reach the next position within the expected time frame, then he might not be able to see the instructed landmark before the end of time limit and thus no 'Yes' prompt would be received even though the user is not disoriented. Since, it is very much possible to be slow in crossing a intersection esp. if it involves clearing the traffic signals, such cases are duly considered and the system inquires whether the intermediate intersection was crossed. If the answer is 'No', then the system delegates the next position to this intermediate intersection and waits for a

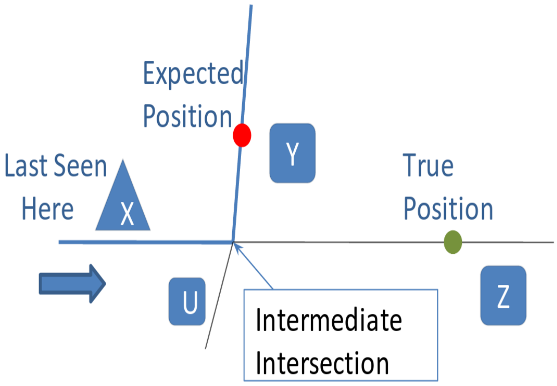

Figure 4.1: **Detecting Disorientation** The highlighted route denotes the path segment communicated to the user for wayfinding. The user has disoriented from the desired path segment and has been detected by the system after a positive response on querying on whether the intermediate intersection was crossed.

'Yes' prompt which if received, confirms that the user has crossed the intersection.

Case-II - Disoriented User

However, if the system receives a 'Yes' prompt at the inquiry for crossing the intersection, it waits for an additional time frame (which is sized as that of the original frame reduced by a wait factor). This additional time frame is used to accomodate possible slowness of the driver assuming that the user though running a little slow would encounter the desired landmark at the next path segment. Once the additional time frame expires and yet the user doesn't see the instructed landmark despite having crossed the intesection, the system traps into the user for reorientation. This situation is depicted in Figure 4.1.

4.5 Reorientation algorithm

The process of tackling disorientation is divided into two phases, each with its own characteristic way of causing reorientation.

Phase-I: Anticipative

The first phase is *anticipative phase* where we estimate user's position based on movement extrapolation using all possible paths from the last seen point (LSP).

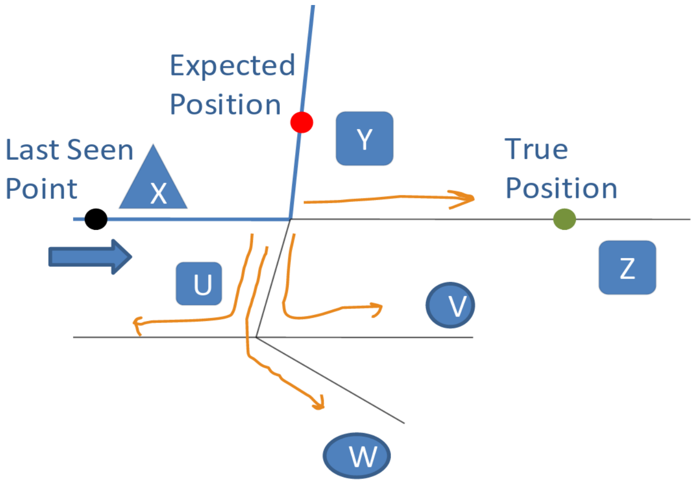

Figure 4.2: **Reorientation:Anticipative-Phase** Prompt the user with questions identifying landmarks *U, V, W* and *Z* along with the orientations w.r.t. user (left or right) to get the location estimate. Reorientation is achieved when the user acknowledges any of the identities put in the prompts.

The idea is to localize the user to the nearest landmark visited by the user. In the act of localization, the user is prompted with the possible landmarks he could have encountered en-route. For example, consider the case shown in Figure 4.2. Once disorientation is detected for the user, then based upon speed predictions for the user, the possible path segments and associated landmarks are identified. Thereafter, we prompt the user with questions identifying these landmarks (*U, V, W* and *Z*) along with the orientations w.r.t. user (left or right) to get the location estimate. Reorientation is achieved when the user acknowledges any of the identities put in the prompts.

Phase-II:Reactive

In a case where small number of prompts suffice in reorientation, one can directly query for identification over all such landmarks seen on the way. But when the number of prompts increases by more than an acceptable extent, we opt to switch the reorientation strategy. Until now, we were being anticipative and the questions asked were of the form '*do you see U on your right?*'. In the reactive strategy, we attempt to capture the environment of the user by querying over well-defined attributes. For example, the questions now are of the form '*do you see any building* *nearby?'* The term '*building*' here is one of the categorizing attributes of landmarks which is a part of the feature-set stored as visual features in the knowledge base[2]. The feature-set includes attributes like color of the buildings, heights, shapes of the open spaces and other automatically extractable attributes.

The major drawback in solely relying upon the above strategy is that it doesn't guarantee to localize a user and is dependent upon the stored visual features. In some cases, even after exhaustively asking questions on the categorizing attributes, there still might be a set of possible locations of the user. For example, in an ideally homogenous neighborhood, buildings in all streets might look alike in shape, color and height. For such cases, when there are still a set of locations under consideration, the user is asked to follow his original direction of movement only to be interrupted later for yet another reorientation. This process is iteratively repeated until the responses resolve the location of the user distinctively.

The reactive approach is motivated by the *scene analysis* location sensing techniques [9]. These techniques work on visual images or electromagnetic measurements to sense observed features of a scene for determining a user's physical location. Understandably, the features used are simple that are easy to represent and compare from an observed scene. The approach mentioned is a hybrid of *static* and *differential* scene analysis. In the former technique, features in question are looked up from a pre-defined geo-spatial database, whereas in the latter, differences in scenes observed with user's movements are used to match known spatial environments.

[2] See Section **??** for more details

Chapter 5

Implementation

In chapter 3, we discussed the architecture of a generic wayfinding model, its structure and interaction of the different modules. In this chapter, the implementation-specific aspects of the key components are described with consideration to its application in developing a location-unaware dialog based system. We discuss a methodology to build the knowledge-base by automated extraction of attribute-values and identification of associations in the entities to represent spatial information. Furthermore, in context of reorientation, we introduce our approach to study movement patterns in order to facilitate early localization of a disoriented user.

5.1 Knowledge Base

A given map dataset can be imported directly into a relational-database model where each shapefile gets converted to an individual relation. The properties of spatial features are converted to attributes and each spatial feature forms a tuple in the relation. The attributes of the spatial features can be divided into four categories:

- **identitificational** - these attributes include *reference-id* as a pointer to the spatial feature, and *category* specifying the class of the spatial feature. The values to category attribute can come directly from the shapefile to which they belong.

- **geometric** - for a spatial database, geometry comes as a primitve datatype. This allows the definition of functions to handle geometry based operations and make several deductions like area, length, etc.
- **visual** - these attributes define the visual attributes like height of a building, color of a playground, etc.
- **semantic** - these attributes are defined as per external sources can meant to identify the meaning behind a feature e.g., petrol-pump, ATM. The major attribute that falls in this category is the popularity of the feature as a landmark which facilitates context-aware wayfinding.

The first two of these attributes can be defined to be *intrinsic* attributes as these are present or can be easily extracted from exisiting spatial databases. While, the other two attributes are *extrinsic*, as external resources need to be used for their extraction.

5.1.1 Extracting Attribute Values

Raubal and Winter [28] worked in a similar research categorising the attributes defining landmarks and provided a methodology to automatically extract local landmarks using integrated datasets such as 3-D city model, navigation graphs and georeferenced images of every house in the neighbourhood. Such an approach offers to improve services giving a with resources on-supply to populate the visual and semantic attributes. But, the research evidently states that to extract visual and semantic features, one needs comprehensive datasets and extrinsic dependence. Brenner and Elias [2] focussed on extracting the landmarks structurally using data mining on spatial databases. The approach focusses more on intrinsic attributes (such as form factor of a building, distance off the road segment,etc.), values of which are extractable on processing from the map itself but also highlight the use of laser methods to identify certain visual features (such as height, visibility). The various attributes considered in their work to extract salient landmarks are shown

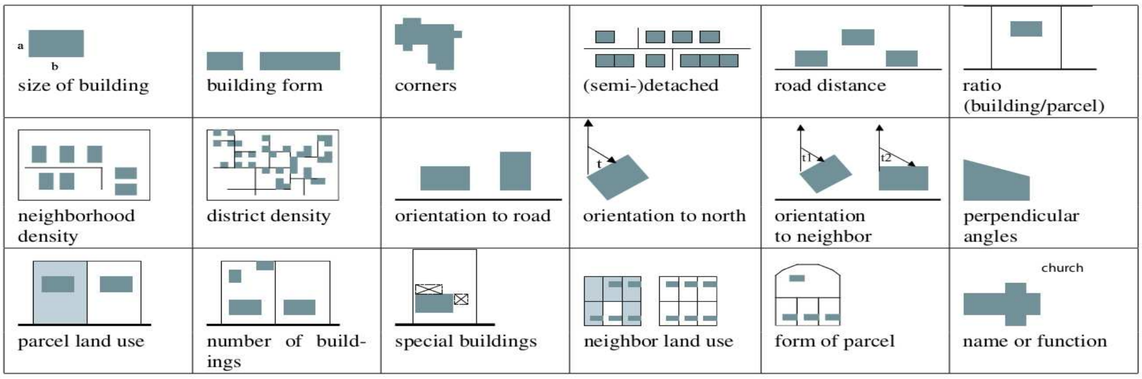

Figure 5.1: **Extracting Geometric Salience** The geometric attributes used by Brenner and Elias [2] to extract building-based landmarks

in Figure 5.1.

We opted to work on the lines similar to [2] to define the geometric salience of buildings. In a real-world application, only those attributes should be considered that are identifiable and communicable. For example, *form of parcel* (in Figure 5.1) might be a difficult attribute to realize in English language and equally difficult to be identified by a driver. On the other hand, *road distance* is easily identifiable and communicable. Apart from the buildings, we also considered open spaces (such as playgrounds or parks) while searching for salient features in a neighborhood. Though a street may well be surrounded by structurally similar buildings but a large enough open-space area in it would be sufficient to confirm orientation of the user if there is no such feature in the neighbouring streets. We do not rely upon the visual and semantic attributes of a landmark for two reasons: (a) the geodatasets used for extracting such information are scarce and limited, and (b) the requirement of our guidance algorithm from the attributes of a landmark is fulfilled if they can make the landmark uniquely identifiable in its local neighborhood. Though visual and semantic attributes would enhance the identifiability but the algorithm is adaptive to the absence of such attributes. We tested the usage of such attributes in guidance and localization using synthetic settings wherein the number of available attributes was parametrized and attribute values were randomly assigned.

Semantic attributes pose a challenge for automated extraction and demand for sophisticated web-mining approaches for qualitative results. Tezuka and Tanaka [32] present one such approach of exploiting world wide web to extract landmarks from digital documents with good precision. However for the purpose of this work, we used a simple and a scalable approach to extract semantic attribute of *popularity* using web resources in order to enhance feature identifiability wherever possible. The popularity of a landmark can help in deducing familiarity of a navigator[1]. For example, if a user comprehends instructions which refer to less popoular landmarks by name, it can be assumed that he is well-familiar with the neighborhood and thus, instead of using geometric features for making a landmark identifiable, the local names of the landmark can be used for wayfinding assistance. The extraction methodology exploits two well-known mapping applications as resources Google maps [26] and Wikimapia [33][2] and is described below. Also, since these applications also collect metadata in the form of user-reviews, they can be utilized to obtain local-names/alias for the spatial features of an environment which are not present in traditional geodatabases.

Using Google Maps

To help provide development services, Google-Maps provides a ranking scheme to search for popular places in nearby area via its Places API. The ranking scheme utilizes the quality of citations and reviews to places done by its extensively large user base. The search results show upto 60 places ranked in the order of prominence for a given query for a given radius. Thus it forms a convenient application to find popular places in a neighborhood or equivalently to find popularity of a given location as compared to its local neighborhood. We applied this methodology to IITK campus and exploited the results of the search to contribute to the *popularity* attribute. Due to a comparatively small area occupied by the campus, the attributes were populated with a single search query with radius set to 2km from PK Kelkar Library, the approximate center of IITK campus.

[1] Also, see Section 4.2

[2] depicted in Figure 5.2

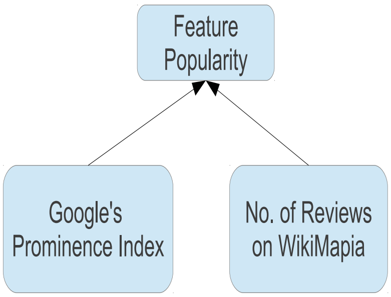

Figure 5.2: **Extracting Popularity** A scalable approach to get an indication on the popularity of a landmark using dependence on mapping applications.

Using Wikimapia

Unlike Google Maps, wikimapia is an open-content mapping application and it provides free access to its extensive geodatabase. The maps in wikimapia are edited and updated via crowdsourcing. And again, unlike Google, it doesn't provide any ranking index to directly find the prominence of a location. But, it does provide an open access to all the reviews submitted for the places in its spatial database. Based on the heuristic that a popular place would have a comparatively large number of reviews submitted, we extract and map popularity to the corresponding locations using the number of reviews as values. Under wikimapia, we define popularity of an area as the total number of reviews submitted of all places enclosed under it.

5.1.2 Identifying associations

Having combined the resources for extracting attribute values in building relations for the features of an environment, the need is to provide an interface to use this spatial information in a real-time wayfinding assistance task. To be able to use the features in route instructions, every road segment needs to be associated with a certain feature (landmark) which serves to confirm the orientation of a user. For the purpose of this work, we used nearness heuristic based on the closest distance between the landmark and the road to identify the association between a landmark and a road segment. The distance threshholds qualifying a landmark's association are chosen to be minimum so as to allow identifiablity of attributes by a driver.

If visibility analysis methods are available (3D models, laser scanning, etc), distance criteria can't be relaxed beyond a certain extent as even though a landmark is visible from the road segment but it can't be guaranteed that the attributes of the landmark can be identified by a distant driver. Also, as per Lynch [22], distant landmarks are used only for overall guidance by novice.

5.2 Facilitating Reorientation

In Section 4.5, we discussed the algorithm to reorient a disoriented user by querying over possible paths. Trajectory predictions can assist in reorientation by studying distraction patterns which can be exploited for faster localization. These patterns may arise due to environmental factors, complexity of the underlying communication protocol or arbitrary human errors. Trajectory prediction algorithms have been previously developed for improving QoS in cellular mobile networks ([18]) and facilitating location based services by predicting future locations ([14]). These situations differ from the context of disorientation in wayfinding in which the movements are biased based upon route instructions.

The approach for our trajectory prediction is bases on probabilistic modelling of movements under the constraints of route-instructions and extends the constraint-free modelling approach adopted in [20]. The model calculates probability of disorienting on a particular road segment (Y) from the given path segments (X, I). To realize the model, the required probability is output from a conditional probability $P(Y/X, I)$ which can be put as

$$D_{S_1,R_1,R_2} = P(Y = S_1/X = R_1, I = R_2)$$

where S_1, R_1 and R_2 are road segments and D_{S_1,R_1,R_2} is the probability to disorient to road segment S_1 when the user is currently on R_1 and instructed to move next to R_2. The probability is estimated based on historical movement patterns and can be formulated as

$$P(Y/X, I) = \frac{N(Y/X, I)}{\sum\limits_{y \in O_{X,I}} N(y/X, I)}$$

where $O_{X,I}$ is the set of all the road segments at the intersection of X and I, and $N(y/X, I)$ is the number of times road segment y has been taken from X when the next instruction was to take I.

Thus, reorientation algorithm sorts the road segments in decreasing order of probability to disorient to, and prompts them incrementally to the user until a positive acknowledgement is received or limit to the number of prompts is exceeded after which it enters into reactive phase.

Chapter 6

Evaluation

This chapter describes an intrinsic evaluation of our proposed model, particularly focussing on the ability to localize and reorient a disoriented user. The evaluation is conducted by modelling user behavior with paramters to characterize speed profile and erroneous behaviour. We then measure the performance of the guidance algorithm under a dialog-based interface with regards to quality of user experience. To our knowledge, there has not been any previous work using a virtual user model for evaluating wayfinding guidance.

6.1 Dataset

We begin by describing the dataset for our prototype implementation. The dataset used for implementation of the model was taken from the geospatial data of Indian Institute of Technology, Kanpur (IITK). The raw data consists of shapefiles representing different layers of a GIS including road network, geometry and metadata of spatial features such as academic buildings, residential area, parks, etc. Visually, the IITK environment depicts a mix of regions which are variably dense in terms of landmark density (Figure 6.1. On one hand, the academic area is filled with a plethora of distinct features which can be easily identified by an unfamiliar navigator and residential regions on the other, have almost no salient landmark or mutually distinguishing features. The academic area is built on a dense street network while

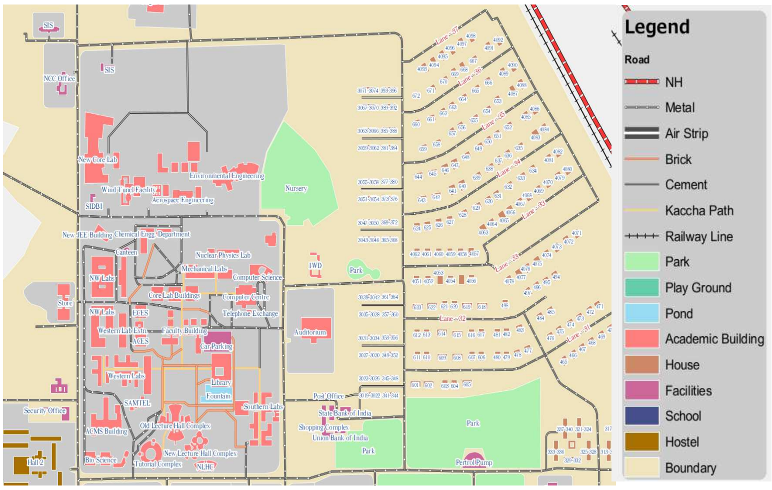

Figure 6.1: **Dataset** Snapshot of the map of Indian Institute of Technology, Kanpur (IITK). The left portion of the snapshot depicting academic buildings (see legend) represents the landmark-dense portion of the campus, Academic area. The organized homogenous space to the right of it represents a portion of the residential areas which has saliently insignificant features.

there are long and clear demarcations in the residential areas [1].

6.2 Simulation Setup

The time spent in executing real-time wayfinding tasks limits the scope of comprehensive testing of a wayfinding-assistance framework. Furthermore, simulations help in situation modelling which has vital importance in studying scenarios of disorientation. The simulation rests on a virtual user model and a synthetic set of feature attributes. The user model simulates varying speed patterns with regards to human behaviour, while the synthetic attribute-set models the extent of diversity in spatial enviroments.

[1] Prior to building the knowledge-base, a preliminary processing is done on the road network data. Since the shapefiles for road network are in raw geo-vector format, they need to be converted to routable network for route computations. The road segments are split at intersections to disjoint edges, each with a start and an end node.

6.2.1 User Modelling

Speed Profiles

The speed of a driver at a particular road segment is modelled using two parameters- 1) *categorical-speed* parameter, to represent the speed preferences of a driver and, 2) *deviation* parameter, to add an element of non-uniformity to speed w.r.t different road segments.

The categorical-speed for the drivers are picked from a continuous probability distribution model. Previous studies ([19, 25]) have found the normal distribution to work under homogenous traffic flow (same type of vehicles) with moderate to low traffic volume conditions. However under heterogenous traffic conditions, researchers have proposed using a lognormal distribution ([7]) for traffic modelling. The lognormal distribution used for sampling categorical-speed is shown in Figure **??**. The parameters of the distributions are set such that 85th percentile speed is equal to the 30 kph i.e. speed limit at IITK campus[2] . The deviation parameter is sampled at each road segment from a normal distribution ($\mu = 0kph$, $\sigma = 5kph$) and added to the categorical-speed to generate observed average speed at the road segment.

Erroneous Behaviour

To cause disorientation in following wayfinding instructions, we modelled a simple error-making probability model of a user which governs user-behaviour at every decision point by choosing each decision with uniform probability. A more sophisticated error-making model would bias decisions based on the structure of the intersection. For instance, one may assign higher probability to go straight when asked to turn left (or right). This may be particularly useful in complex intersections where it maybe difficult to follow route instructions and comprehend in a degenerate natural-language translation of the clock-based convention model.

[2] This is in accordance with the observed relationship between vehicle operating speeds and posted speed limits.

6.2.2 Synthetic Attributes

We had discussed possible approaches to extract attributes of a spatial feature in Section 5.1. Although since the focus of this work was on dialog-based localization, we chose to synthetically populate the attribute-set of the features. The number of attributes was paramterized in the testbed and the corresponding values were assigned randomly from a fixed set. The idea was to visualize dependency of performance of the localization algorithm on the presence of extrinsic attributes.

6.3 Goodness Metric

The major concern with any wayfinding assistance is the extent of interference it causes to the wayfinder. Thus, quality of user experience in a dialog-based wayfinding system would have major dependence on the number of prompts made while assisting to the user. As the complexity of routes increase in terms of number of decisions to be made en-route, greater number of prompts have to be put to the user. Also, if the user commits an error of judgement in following the route instructions and disorients from the intended path, additional prompts need to be put in order to reorient the user.

Based on the above discussion, we define *goodness* for the evaluation of the proposed model with the metric as number of prompts made to the user (N_p) normalized over the total number of decisions made (N_D) and, the total number of disorientations (N_e) recorded before reaching the destination. The goodness metric (G) is formulated as:

$$G = \frac{N_p - \alpha \times N_D}{N_e}$$

where α is a constant, representing the number of prompts made in guiding a user on a route with exactly one decision point.

6.4 Results

Chapter 7

Conclusion and Future Work

Bibliography

[1] Jörg Baus, Antonio Krüger, and Wolfgang Wahlster. A resource-adaptive mobile navigation system. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 15–22. ACM, 2002.

[2] C Brenner and B Elias. Extracting landmarks for car navigation systems using existing gis databases and laser scanning. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W8):131–138, 2003.

[3] Heriberto Cuayáhuitl, Nina Dethlefs, Kai-Florian Richter, Thora Tenbrink, and John Bateman. A dialogue system for indoor wayfinding using text-based natural language. *International Journal of Computational Linguistics and Applications, ISSN 0976*, 962, 2010.

[4] Robert Dale and Sabine Geldof. Using natural language generation in automatic route description. In *Journal of Research and Practice in Information Technology*, pages 89–105, 2005.

[5] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3):143–166, 2003.

[6] Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. The give-2 corpus of giving instructions in virtual environments. In *LREC*, 2010.

[7] Daniel L Gerlough and Matthew J Huber. *Traffic flow theory: a monograph*, volume 165. Transportation Research Board, National Research Council Washington, DC, 1975.

[8] Christopher Habel. Incremental generation of multimodal route instructions. In *Working Papers of the 2003 AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, pages 44–51, 2003.

[9] Jeffrey Hightower and Gaetano Borriello. Location sensing techniques. *IEEE Computer*, 34(8):57–66, 2001.

[10] Topi Hurtig and Kristiina Jokinen. Modality fusion in a route navigation system. In *Procs of the Workshop on Effective Multimodal Dialogue Interfaces EMMDI-2006. Sydney, Australia*, 2006.

[11] Srini Janarthanam and Oliver Lemon. The gruve challenge: generating routes under uncertainty in virtual environments. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 208–211. Association for Computational Linguistics, 2011.

[12] Srinivasan Janarthanam, Oliver Lemon, Phil Bartie, Tiphaine Dalmas, Anna Dickinson, Xingkun Liu, William Mackaness, and Bonnie Webber. Evaluating a city exploration dialogue system combining question-answering and pedestrian navigation. In *Proc. ACL*, 2013.

[13] Kristiina Jokinen, Antti Kerminen, Mauri Kaipainen, Tommi Jauhiainen, Graham Wilcock, Markku Turunen, Jaakko Hakulinen, Jukka Kuusisto, and Krista Lagus. Adaptive dialogue systems-interaction with interact. In *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue-Volume 2*, pages 64–73. Association for Computational Linguistics, 2002.

[14] Hassan A. Karimi and Xiong Liu. A predictive location model for location-based services. In *Proceedings of the 11th ACM International Symposium on*

Advances in Geographic Information Systems, GIS '03, pages 126–133, New York, NY, USA, 2003. ACM.

[15] Alexander Klippel. Wayfinding choremes. In *Spatial Information Theory. Foundations of Geographic Information Science*, pages 301–315. Springer, 2003.

[16] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *5th ACM/IEEE International Conference on Human-Robot Interaction*, pages 259–266, 2010.

[17] Parisa Kordjamshidi, Martijn Van Otterlo, and Marie-Francine Moens. Spatial role labeling: Towards extraction of spatial relations from natural language. *ACM Transactions on Speech and Language Processing*, 8(3):4:1–4:36, 2011.

[18] Miltiadis Kyriakakos, Stathes Hadjiefthymiades, Nikolaos Frangiadakis, and L Merakos. Multi-user driven path prediction algorithm for mobile computing. In *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pages 191–195. IEEE, 2003.

[19] HJ Leong. The distribution and trend of free speeds on two lane two way rural highways in new south wales. In *Australian Road Research Board (ARRB) Conference, 4th, 1968, Melbourne*, volume 4, 1968.

[20] Xiong Liu and Hassan A Karimi. Location awareness through trajectory prediction. *Computers, Environment and Urban Systems*, 30(6):741–756, 2006.

[21] Kristin L. Lovelace, Mary Hegarty, and Daniel R. Montello. Elements of good route directions in familiar and unfamiliar environments. In *Proceedings of the International Conference on Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science*, COSIT '99, pages 65–82, London, UK, UK, 1999. Springer-Verlag.

[22] Kevin Lynch. *The image of the city*, volume 11. MIT press, 1960.

[23] Wolfgang Maaß, Jörg Baus, and Joachim Paul. *Visual grounding of route descriptions in dynamic environments*. Univ. des Saarlandes, SFB 314, 1995.

[24] Cynthia Matuszek, Dieter Fox, and Karl Koscher. Following directions using statistical machine translation. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, pages 251–258. IEEE Press, 2010.

[25] JR McLean. Observed speed distributions and rural road traffic operations. In *Australian Road Research Board Conference Proc*, volume 9, 1979.

[26] Map of Mumbai. Google maps, https://maps.google.com/, May 2014.

[27] Thiyagarajasarma Pattabhiraman and Nick Cercone. Selection: Salience, relevance and the coupling between domain-level tasks and text planning. In *Proceedings of the Fifth International Workshop on Natural Language Generation, Dawson, PA*, pages 79–86, 1990.

[28] Martin Raubal and Stephan Winter. *Enriching wayfinding instructions with local landmarks*. Springer, 2002.

[29] Kai-Florian Richter, Martin Tomko, and Stephan Winter. A dialog-driven process of generating route directions. *Computers, Environment and Urban Systems*, 32(3):233–245, 2008.

[30] Philipp Spanger, Ryu Iida, Takenobu Tokunaga, Asuka Terai, and Naoko Kuriyama. A task-performance evaluation of referring expressions in situated collaborative task dialogues. *Language Resources and Evaluation*, 47(4):1285–1304, 2013.

[31] Raghavan Srinivasan and Paul P Jovanis. Effect of selected in-vehicle route guidance systems on driver reaction times. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 39(2):200–215, 1997.

[32] Taro Tezuka and Katsumi Tanaka. Landmark extraction: A web mining approach. In *Spatial information theory*, pages 379–396. Springer, 2005.

[33] Wikimapia. http://wikimapia.org. Accessed May 4, 2014.

[34] R. Michael Young. Using grice's maxim of quantity to select the content of plan descriptions. *Artificial Intelligence*, 115:215–256, 1999.