

A Web-based Evaluation Framework for Spatial Instruction-Giving Systems

Srinivasan Janarthanam, Oliver Lemon, and Xingkun Liu

Interaction Lab

School of Mathematical and Computer Sciences

Heriot Watt University, Edinburgh

sc445, o.lemon, x.liu@hw.ac.uk

Abstract

We demonstrate a web-based environment for development and testing of different pedestrian route instruction-giving systems. The environment contains a City Model, a TTS interface, a game-world, and a user GUI including a simulated street-view. We describe the environment and components, the metrics that can be used for the evaluation of pedestrian route instruction-giving systems, and the shared challenge which is being organised using this environment.

1 Introduction

Generating navigation instructions in the real world for pedestrians is an interesting research problem for researchers in both computational linguistics and geo-informatics (Dale et al., 2003; Richter and Duckham, 2008). These systems generate verbal route directions for users to go from A to B, and techniques range from giving ‘a priori’ route directions (i.e. all route information in a single turn) and incremental ‘in-situ’ instructions, to full interactive dialogue systems (see section 4). One of the major problems in developing such systems is in evaluating them with real users in the real world. Such evaluations are expensive, time consuming and painstaking to organise, and are carried out not just at the end of the project but also during the development cycle. Consequently, there is a need for a common platform to effectively compare the performances of verbal navigation systems developed by different teams using a variety of techniques (e.g. a priori vs. in-situ or rule-based vs. machine learning).

This demonstration system brings together existing online data resources and software toolkits to create a low-cost framework for evaluation of pedestrian route instruction systems. We have built a web-based environment containing a *simulated real world* in which users can simulate walking on the streets of real cities whilst interacting with different navigation systems. This evaluation framework will be used in the near future to evaluate a series of instruction-giving dialogue systems.

2 Related work

The GIVE challenge developed a 3D virtual indoor environment for development and evaluation of indoor pedestrian navigation instruction systems (Koller et al., 2007; Byron et al., 2007). In this framework, users can walk through a building with rooms and corridors, similar to a first-person shooter game. The user is instructed by a navigation system that generates route instructions. The basic idea was to have several such navigation systems hosted on the GIVE server and evaluate them in the same game worlds, with a number of users over the internet. Conceptually our work is very similar to the GIVE framework, but its objective is to evaluate systems that instruct pedestrian users in the real world. The GIVE framework has been successfully used for comparative evaluation of several systems generating instructions in virtual indoor environments.

Another system, “Virtual Navigator”, is a simulated 3D environment that simulates the real world for training blind and visually impaired people to learn often-used routes and develop basic navigation skills (McGookin et al., 2010). The framework

uses haptic force-feedback and spatialised auditory feedback to simulate the interaction between users and the environment they are in. The users simulate walking by using arrow keys on a keyboard and by using a device that works as a 3D mouse to simulate a virtual white cane. Auditory clues are provided to the cane user to indicate for example the difference between rush hour and a quiet evening in the environment. While this simulated environment focusses on the providing the right kind of tactile and auditory feedback to its users, we focus on providing a simulated environment where people can look at landmarks and navigate based on spatial and visual instructions provided to them.

User simulation modules are usually developed to train and test reinforcement learning based interactive spoken dialogue systems (Janarthanam and Lemon, 2009; Georgila et al., 2006; Schatzmann et al., 2006). These agents replace real users in interaction with dialogue systems. However, these models simulate the users' behaviours in addition to the environment in which they operate. Users' dialogue and physical behaviour are dependent on a number of factors such as a user's preferences, goals, knowledge of the environment, environmental constraints, etc. Simulating a user's behaviour realistically based on many such features requires large amounts of data. In contrast to this approach, we propose a system where only the spatial and visual environment is simulated.

See section 4 for a discussion of different pedestrian navigation systems.

3 Architecture

The evaluation framework architecture is shown in figure 1. The server side consists of a broker module, navigation system, gameworld server, TTS engine, and a city model. On the user's side is a web-based client that consists of the simulated real-world and the interaction panel.

3.1 Game-world module

Walking aimlessly in the simulated real world can be a boring task. Therefore, instead of giving web users navigation tasks from A to B, we embed navigation tasks in a game-world overlaid on top of the simulated real world. We developed a "treasure hunting"

game which consists of users solving several pieces of a puzzle to discover the location of the treasure chest. In order to solve the puzzle, they interact with game characters (e.g. a pirate) to obtain clues as to where the next clue is. This sets the user a number of navigation tasks to acquire the next clues until they find the treasure. In order to keep the game interesting, the user's energy depletes as time goes on and they therefore have limited time to find the treasure. Finally, the user's performance is scored to encourage users to return. The game characters and entities like keys, chests, etc. are laid out on real streets making it easy to develop a game without developing a game-world. New game-worlds can be easily scripted using Javascript, where the location (latitude and longitude) and behaviour of the game characters are defined. The game-world module serves game-world specifications to the web-based client.

3.2 Broker

The broker module is a web server that connects the web clients to their corresponding different navigation systems. This module ensures that the framework works for multiple users. Navigation systems are instantiated and assigned to new users when they first connect to the broker. Subsequent messages from the users will be routed to the assigned navigation system. The broker communicates with the navigation systems via a communication platform thereby ensuring that different navigation systems developed using different languages (such as C++, Java, Python, etc) are supported.

3.3 Navigation system

The navigation system is the central component of this architecture, which provides the user instructions to reach their destinations. Each navigation system is run as a server remotely. When a user's client connects to the server, it instantiates a navigation system object and assigns it to the user exclusively. Every user is identified using a unique id (UUID), which is used to map the user to his/her respective navigation system. The navigation system is introduced in the game scenario as a buddy system that will help the user in his objective: find the treasure. The web client sends the user's location to the system periodically (every few seconds).

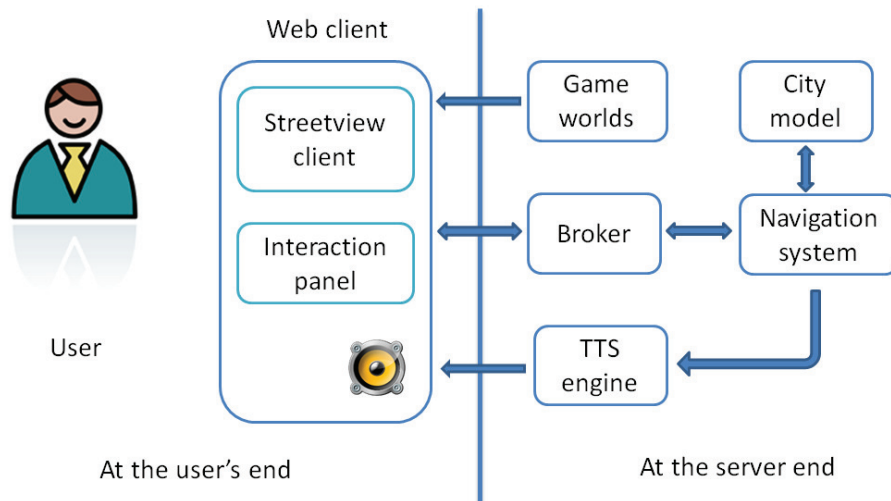


Figure 1: Evaluation framework architecture

3.4 TTS engine

Alongside the navigation system we use the Cereproc text-to-speech engine that converts the utterances of the system into speech. The URL of the audio file is then sent to the client's browser which then uses the audio plugin to play the synthesized speech to the user. The TTS engine need not be used if the output modality of the system is just text.

3.5 City Model

The navigation system is supported by a database called the City Model. The City Model is a GIS database containing a variety of data required to support navigation tasks. It has been derived from an open-source data source called OpenStreetMaps¹. It consists of the following:

- Street network data: the street network data consists of nodes and ways representing junctions and streets.
- Amenities: such as ATMs, public toilets, etc.
- Landmarks: other structures that can serve as landmarks. E.g. churches, restaurants, etc.

The amenities and landmarks are represented as nodes (with latitude and longitude information). The City Model interface API consists of a number of

¹www.openstreetmaps.org

subroutines to access the required information such as the nearest amenity, distance or route from A to B, etc. These subroutines provide the interface between the navigation systems and the database.

3.6 Web-based client

The web-based client is a JavaScript/HTML program running on the user's web browser software (e.g. Google Chrome). A snapshot of the webclient is shown in figure 2. It has two parts: the streetview panel and the interaction panel.

Streetview panel: the streetview panel presents a simulated real world visually to the user. When the page loads, a Google Streetview client (Google Maps API) is created with an initial user coordinate. Google Streetview is a web service that renders a panoramic view of real streets in major cities around the world. This client allows the web user to get a panoramic view of the streets around the user's virtual location. A gameworld received from the server is overlaid on the simulated real world. The user can walk around and interact with game characters using the arrow keys on his keyboard or the mouse. As the user walks around, his location (stored in the form of latitude and longitude coordinates) gets updated locally. Streetview also returns the user's point of view (0-360 degrees), which is also stored locally.

Interaction panel: the web-client also includes an

interaction panel that lets the user interact with his buddy navigation system. In addition to user location information, users can also interact with the navigation system using textual utterances or their equivalents. We provide users with two types of interaction panel: a GUI panel and a text panel. In the GUI panel, there are GUI objects such as buttons, drop-down lists, etc. which can be used to construct requests and responses to the system. By clicking the buttons, users can send abstract semantic representations (dialogue actions) that are equivalent to their textual utterances. For example, the user can request a route to a destination by selecting the street name from a drop down list and click on the Send button. Similarly, users can click on ‘Yes’, ‘No’, ‘OK’, etc. buttons to respond to the system’s questions and instructions. In the text panel, on the other hand, users are free to type any request or response they want. Of course, both types of inputs are parsed by the navigation system. We also plan to add an additional input channel that can stream user speech to the navigation system in the future.

4 Candidate Navigation Systems

This framework can be used to evaluate a variety of navigation systems. Route navigation has been an interesting research topic for researchers in both geoinformatics and computational linguistics alike. Several navigation prototype systems have been developed over the years. Although there are several systems that do not use language as a means of communication for navigation tasks (instead using geo-tagged photographs (Beeharee and Steed, 2006; Hiley et al., 2008), haptics (Bosman et al., 2003), music (Holland et al., 2002; Jones et al., 2008), etc), we focus on systems that generate instructions in natural language. Therefore, our framework does not include systems that generate routes on 2D/3D maps as navigation aids.

Systems that generate text/speech can be further classified as follows:

- ‘A priori’ systems: these systems generate route instructions prior to the users touring the route. These systems describe the entire route before the user starts navigating. Several web services exist that generate such lists of step-by-step instructions (e.g. Google/Bing direc-

tions).

- ‘In-situ’ or incremental route instruction systems: these systems generate route instructions incrementally along the route. e.g. CORAL (Dale et al., 2003). They keep track of the user’s location and issue the next instruction when the user reaches the next node on the planned route. The next instruction tells the user how to reach the new next node. Some systems do not keep track of the user, but require the user to request the next instruction when they reach the next node.
- Interactive navigation systems: these systems are both incremental and interactive. e.g. DeepMap (Malaka and Zipf, 2000). These systems keep track of the user’s location and proactively generate instructions based on user proximity to the next node. In addition, they can interact with users by asking them questions about entities in their viewshed. For example “Can you see a tower at about 100 feet away?”. Questions like these will let the system assess the user’s location and thereby adapt its instruction to the situated context.

5 Evaluation metrics

Navigation systems can be evaluated using two kinds of metrics using this framework. Objective metrics such as *time taken* by the user to finish each navigation task and the game, *distance travelled*, number of *wrong turns*, etc. can be directly measured from the environment. Subjective metrics based on each user’s ratings of different features of the system can be obtained through user satisfaction questionnaires. In our framework, users are requested to fill in a questionnaire at the end of the game. The questionnaire consists of questions about the game, the buddy, and the user himself, for example:

- Was the game engaging?
- Would you play it again (i.e. another similar gameworld)?
- Did your buddy help you enough?



Figure 2: Snapshot of the web client

- Were the buddy instructions easy to understand?
- Were the buddy instructions ever wrong or misplaced?
- If you had the chance, will you choose the same buddy in the next game?
- How well did you know the neighbourhood of the gameworld before the game?

6 Evaluation scenarios

We aim to evaluate navigation systems under a variety of scenarios.

- Uncertain GPS: GPS positioning available in smartphones is erroneous (Zandbergen and Barbeau, 2011). Therefore, one scenario for evaluation would be to test how robustly navigation systems handle erroneous GPS signals from the user's end.
- Output modalities: the output of navigation systems can be presented in two modalities: text and speech. While speech may enable a hands-free eyes-free navigation, text displayed on navigation aids like smartphones may increase cognitive load. We therefore believe it

will be interesting to evaluate the systems in both conditions and compare the results.

- Noise in user speech: for systems that take as input user speech, it is important to handle noise in such a channel. Noise due to wind and traffic is most common in pedestrian scenarios. Scenarios with different levels of noise settings can be evaluated.
- Adaptation to users: returning users may have learned the layout of the game world. An interesting scenario is to examine how navigation systems adapt to user's increasing spatial and visual knowledge.

Errors in GPS positioning of the user and noise in user speech can be simulated at the server end, thereby creating a range of challenging scenarios to evaluate the robustness of the systems.

7 The Shared Challenge

We plan to organise a shared challenge for outdoor pedestrian route instruction generation, in which a variety of systems can be evaluated. Participating research teams will be able to use our interfaces and modules to develop navigation systems. Each team will be provided with a development toolkit

and documentation to setup the framework in their local premises for development purposes. Developed systems will be hosted on our challenge server and a web based evaluation will be organised in consultation with the research community (Janarthanam and Lemon, 2011).

8 Demonstration system

At the demonstration, we will present the evaluation framework along with a demo navigation dialogue system. The web-based client will run on a laptop using a high-speed broadband connection. The navigation system and other server modules will run on a remote server.

Acknowledgments

The research has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216594 (SPACEBOOK project www.spacebookproject.org).

References

- Ashweeni K. Beeharee and Anthony Steed. 2006. A natural wayfinding exploiting photos in pedestrian navigation systems. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services (2006)*.
- S. Bosman, B. Groenendaal, J. W. Findlater, T. Visser, M. de Graaf, and Panos Markopoulos. 2003. GentleGuide: An Exploration of Haptic Output for Indoors Pedestrian Guidance. In *Proceedings of 5th International Symposium, Mobile HCI 2003, Udine, Italy*.
- D. Byron, A. Koller, J. Oberlander, L. Stoia, and K. Striegnitz. 2007. Generating Instructions in Virtual Environments (GIVE): A challenge and evaluation testbed for NLG. In *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*.
- Robert Dale, Sabine Geldof, and Jean-Philippe Prost. 2003. CORAL : Using Natural Language Generation for Navigational Assistance. In *Proceedings of the Twenty-Sixth Australasian Computer Science Conference (ACSC2003), 4th7th February, Adelaide, South Australia*.
- Kallirroi Georgila, James Henderson, and Oliver Lemon. 2006. User simulation for spoken dialogue systems: Learning and evaluation. In *Proceedings of Inter-speech/ICSLP*, pages 1065–1068.
- Harlan Hiley, Ramakrishna Vedantham, Gregory Cuelar, Alan Liuy, Natasha Gelfand, Radek Grzeszczuk, and Gaetano Borriello. 2008. Landmark-based pedestrian navigation from collections of geotagged photos. In *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia (MUM) 2008*.
- S. Holland, D. Morse, and H. Gedenryd. 2002. Audio-gps: Spatial audio navigation with a minimal attention interface. *Personal and Ubiquitous Computing*, 6(4):253–259.
- Srini Janarthanam and Oliver Lemon. 2009. A User Simulation Model for learning Lexical Alignment Policies in Spoken Dialogue Systems. In *European Workshop on Natural Language Generation*.
- Srini Janarthanam and Oliver Lemon. 2011. The GRUVE Challenge: Generating Routes under Uncertainty in Virtual Environments. In *Proceedings of ENLG / Generation Challenges*.
- M. Jones, S. Jones, G. Bradley, N. Warren, D. Bainbridge, and G. Holmes. 2008. Ontrack: Dynamically adapting music playback to support navigation. *Personal and Ubiquitous Computing*, 12(7):513–525.
- A. Koller, J. Moore, B. Eugenio, J. Lester, L. Stoia, D. Byron, J. Oberlander, and K. Striegnitz. 2007. Shared Task Proposal: Instruction Giving in Virtual Worlds. In *Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*.
- Rainer Malaka and Er Zipf. 2000. Deep Map - challenging IT research in the framework of a tourist information system. In *Information and Communication Technologies in Tourism 2000*, pages 15–27. Springer.
- D. McGookin, R. Cole, and S. Brewster. 2010. Virtual navigator: Developing a simulator for independent route learning. In *Proceedings of Workshop on Haptic Audio Interaction Design 2010, Denmark*.
- Kai-Florian Richter and Matt Duckham. 2008. Simplest instructions: Finding easy-to-describe routes for navigation. In *Proceedings of the 5th international conference on Geographic Information Science*.
- Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21:97–126.
- P. A. Zandbergen and S. J. Barbeau. 2011. Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. *Journal of Navigation*, 64(3):381–399.