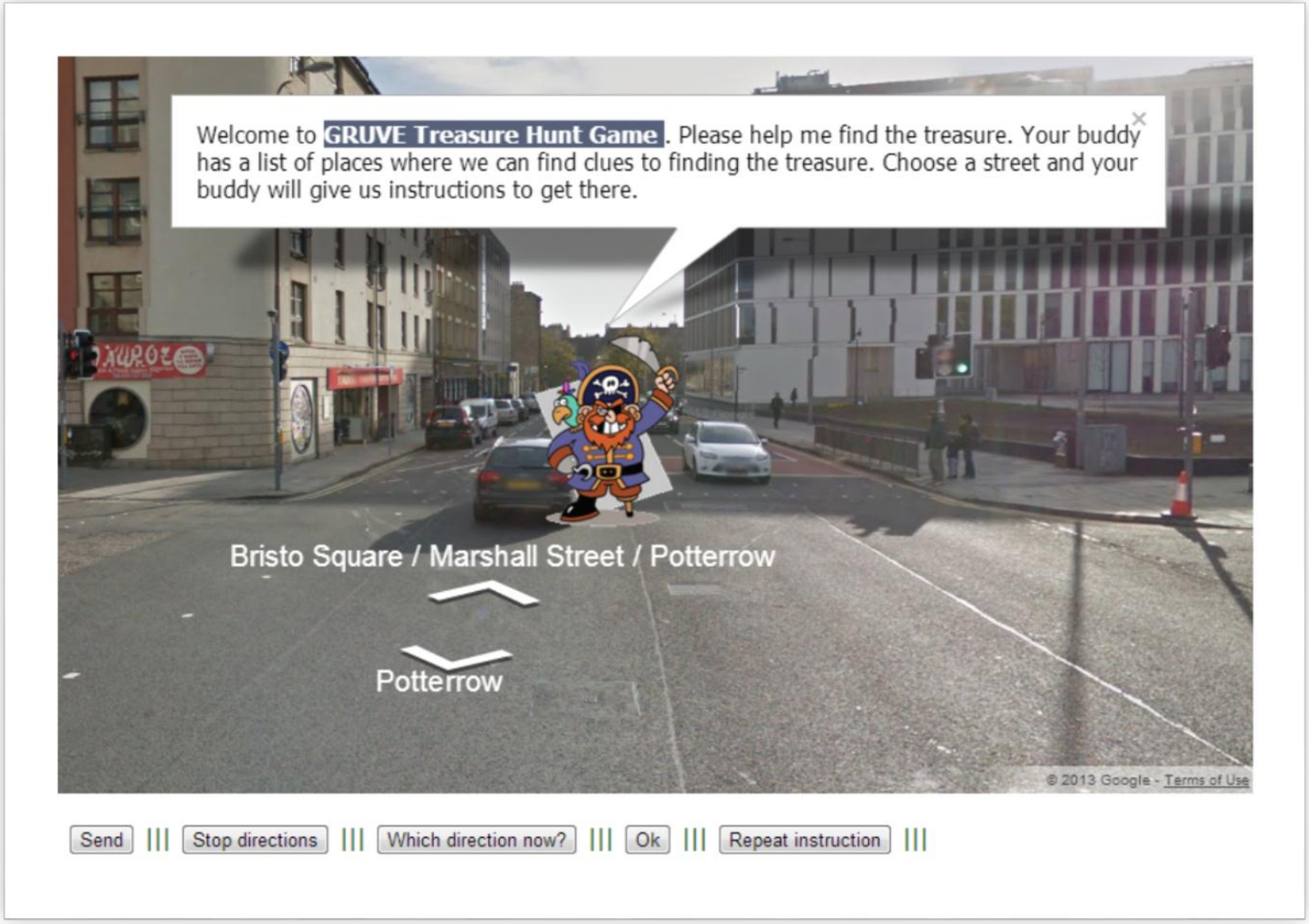

Welcome to GRUVE Treasure Hunt Game. Please help me find the treasure. Your buddy has a list of places where we can find clues to finding the treasure. Choose a street and your buddy will give us instructions to get there.

Bristo Square / Marshall Street / Potterrow
Potterrow
© 2013 Google - Terms of Use
Send
Stop directions
Which direction now?
Ok
Repeat instruction

Giving Route Instructions in Uncertain Virtual Environments

PROJECT SPECIFICATIONS

Saravanan Ganesh | CS599 Project | April 9, 2013

Introduction

Generating navigation instructions in the real world for pedestrians is an interesting research problem for researchers in both computational linguistics and geo-informatics. These systems generate verbal route directions for users to go from A to B, and techniques range from giving ‘a priori’ route directions and incremental ‘in-situ’ instructions, to full interactive dialogue systems.

The project aims to build a Natural Language Generation (NLG) ‘module’ that operates as a part of the bigger interactive dialogue system that assists a user navigating through the streets of a city on Google Streetview. The challenge here is to generate instructions in the context of uncertain user positioning information and to adapt them to a user’s experience in the neighborhood.

The NLG module will fit into a bigger framework which together makes a dialogue system called ‘user buddy’. This dialogue system is available for help for users in an online game. The dialogue system will receive navigation requests from the user. Given information about the user current position, orientation, the path to the destination, the city, the user history, the module suggests steps to be taken, either as one instruction for the next move or instructions for all the moves to reach the destination. The NLG module uses the city model, route plan, and user model to access information about the street layout, entities on the streets and user’s experience in the neighborhood. Using this information, an instruction is generated whenever required.

The framework that hosts the NLG module is built as 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 project draws inspiration from the lectures on Natural Language Generation and Dialogue System Evaluation: those presented during Weeks 5 and 6 in the course CS599.

Example Data

Input and Output data are discussed as applicable to the NLG module.

Input data, is available as a set of Java Class objects and JSON Objects. The City model and User model, as Java class objects are passed as parameters while constructing the NLG module, and other information pertaining to the user's current state are passed as a JSON Object while calling the function.

The input data can be presented as a list of available objects:

- cityModel – Java class with a list of Nodes, Ways, and Entities.
- userModel – Java class with previous Routes, Node visits, Profile and Session Information, Current Route
- route – Java class with Distance to Destination, and Nodes on the way.
- userOrientation – Double member storing current user orientation.
- userCoordinates – Java class storing current user coordinates.

The system presents navigation instruction when asked for. Thus, the output data is always 'one sentence informing the user, her next move towards the goal'.

Examples of these sentences in the baseline system include:

- Continue walking onto "street_name".
- Turn slightly right onto "street_name".
- Turn right and walk onto "street_name" etc.

Resources

The project aims at building the NLG module for a spatial instruction giving system. This system was developed as a framework to facilitate testing of different NLG modules, in a game world scenario.

The framework comes with components that can be used while building the NLG module. These components are user interfaces for information that is available to the module which performs NLG. These components and their uses are outlined below:

1. GRUVE User Interface:

   The game is presented to the user on the GRUVE user interface. The user interface consists of two panels: the gameworld panel (using Google Streetview) and the interaction panel. The game characters are overlaid on top of the real world simulation in the gameworld panel. Users can walk through the streets by clicking on the arrow heads shown on the simulation or using the arrow keys on their keyboard. Users' location (i.e. latitude and longitude) information is sent to the dialogue system as they move around. Using the buttons and dropdown lists, users can interact with the dialogue system to communicate their navigation goals.

2. Wizard’s Desk

   The wizard's desk is a Wizard-of-Oz tool to simulate the dialogue system's behavior. This connects to the GRUVE user interface via a Java servlet. Users' communication via the interaction panel are sent to the wizard's desk for which the human wizard can respond. The response is delivered to the user playing the game the same way the responses of the dialogue system are delivered.

3. Map Viewer

   The Map viewer tool can be used to visualize the underlying street network overlaid on the map of Edinburgh. The network is displayed as a graph with nodes (red marker 📍) representing street junctions and black lines representing streets. The map also presents the amenities such as shops, traffic lights, etc. that can be used by the NLG system as landmarks (using a grey marker 📍) on the map.

4. Log Viewer

   The Log viewer is where the interaction between the user and buddy are stored.

5. Buddy Dialogue System

   The buddy dialogue system is implemented as a set of Java files. The dialogue system and the user interface interact using a java servlet.

Project Plan

The Natural Language Generation module that has to be implemented for the GRUVE challenge, is implemented as a class called HWUNLG. This class has two member variables: HWCityModel and UserModel, and one member function: presentRoute, with a JSON Object parameter. Most of the application logic of the project will deal with these inputs to build an instruction that has to be presented to the user. The following may be considered milestones in the project plan:

1) Build a baseline system using a decision tree algorithm, to choose a custom instruction with values from the available information.
2) Collect instructions from human beings in the game scenario, instead of using the NLG module. This will serve as a corpus for training and testing the system. These instructions will be collected from friends and volunteers.
3) Note features from the information available and study how they vary with instructions from humans.
4) Build a set of all instructions as text, and survey as to find what makes a useful and interesting instruction.
5) Draw a structure for all instructions, from the human instructions and survey, and attempt to list domain values for each part of that structure.
6) Train models to build instruction structure and models to fill the structure.
7) Devise post processing methods for the instruction to make them complete and pleasant.

The plan outlines two significant methods to generate instructions: parameters to instruction mapping and model backed frame based generation.

Pending Issues

The GRUVE user interface lacks quality of an elegant user interface. It may come in the way of the user experience while she tests the generation module. It is necessary to investigate ways to fine tune it and contribute to a pleasant user experience. As the source of the framework is available this seems to be possible, but further investigation is necessary to find what is allowed in the official challenge.

Evaluation

The final evaluation will be performed by the group conducting the GRUVE challenge. As mentioned in their paper – "Navigation systems can be evaluated using two kinds of metrics. 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 measure from the environment. Subjective metrics based on each user's ratings of different features of the system can be obtained through user satisfaction questionnaires. The questionnaire consists of questions about the game, the buddy, and the user himself, for example:

- Was the game engaging?
- Did your buddy help you enough?
- Were the buddy instructions easy to understand?
- If you have the chance, will you choose the same buddy in the next game?
- How well did you know the neighborhood of the gameworld before the game?"
- Would you play it again?

Nevertheless, pre-event testing will be done similar to as outlined in the paper to prepare the system for evaluation. Subjective metrics will be given more emphasis, since the overall aim of the system is to be an intelligent user buddy.

References

Srinivasan Janarthanam, Oliver Lemon, and Xingkun Liu, 2012. A Web-based Evaluation Framework for Spatial Instruction-Giving Systems, In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics.