SpeechCity: A Conversational City Guide based on Open Data

Verena Rieser, Srinivasan Janarthanam, Andy Taylor, Yanchao Yu and Oliver Lemon
Interaction Lab
Heriot-Watt University
Edinburgh, UK
Email: `v.t.rieser@hw.ac.uk`; Website: `www.macs.hw.ac.uk/InteractionLab`

Abstract

We demonstrate a mobile application that assists users in planning a day out within a urban environment. Locality-specific information is acquired from open data sources, and can be accessed via intelligent interaction. For demonstrating the full functionality of the system, we simulate a (user-specified) walking route within the city of Edinburgh, where the system "pushes" relevant information to the user. Through the use of open data, the agent is easily portable and extendable to new locations and domains, which we plan to demonstrate in our future work.

1 Introduction

This project seeks to convert research prototypes of spatial interaction components (Janarthanam et al., 2013), into a robust and extendable mobile application. In particular, we aim to produce a technology which is easily portable to new locations and domains. As such, the core technology of this project operates over open data sources which are scraped from the web to create a "CityModel" (Section 2.2). The response time of the system is improved using a client-server architecture, as described in Section 2.1. We also developed a modular, multi-threaded interaction manager where individual modules are transferrable to new domains (Section 2.3). In contrast to existing mobile applications, our Android agent is able to simultaneously interleave multiple tasks, e.g. navigation and tourist information. The system can also take initiative and push relevant information to the user, based on task priority and user interests (Section 3). In this demo, we present a system for tourist information within the city of Edinburgh. The example dialogue in Table 1 illustrates the system's capabilities on navigation, search, weather, events and information push on Points of Interests (PoI).

User:	Take me to the National Museum.
System:	Ok. Here is a route on the map. [*displays map*]
User:	Thanks. Will it rain today?
System:	Showers are forecasted throughout the day.
User:	Oh no. Is there a cafe nearby?
System:	There are several cafes close to you. The closest one is the Elephant House. [*displays locations on the map*]
User:	Is it any good?
System:	It has a 4-star rating on Foursquare.
System:	It also features a literary pub quiz at 6pm today.
User:	[*continues walking*]
System:	On your right you can see a statue of a dog known as Greyfriars Bobby. Do you want me to tell you more?

Table 1: An example SpeechCity interaction.

2 Architecture

The architecture of the system is shown in Figure 1. The application adopts a client-server architecture. The Android mobile app serves as a client and consists of the interface and dialogue system. The server consists of the CityModel.

2.1 Android Interface

The Android interface module manages the GUI display and user input, and serves as the interface between the user and the dialogue system. It accesses the onboard sensors to determine user's positioning and pace and uses Google ASR API for recognising user speech. Th information are sent to the dialogue system as user inputs. It receives dialogue system outputs such as system utterances and information to be displayed on the screen as well as on the map. The interface uses onboard Google TTS for speech synthesis. It uses OpenStreetMaps API to display the map with user location and other information such as location of restaurants, pubs, etc that the user requested. In addition to spoken interaction, the system also features multi-modal interaction for the user to query the system from a list displayed or by clicking on the map, see Figure 2.

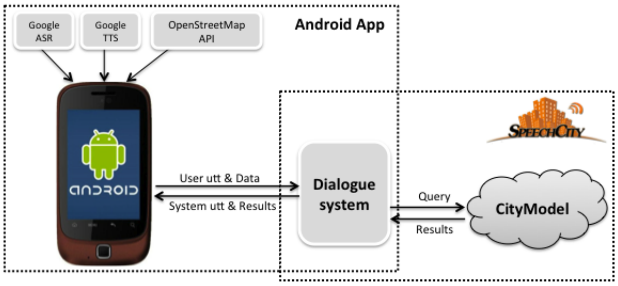

Figure 1: Architecture of SpeechCity mobile application.

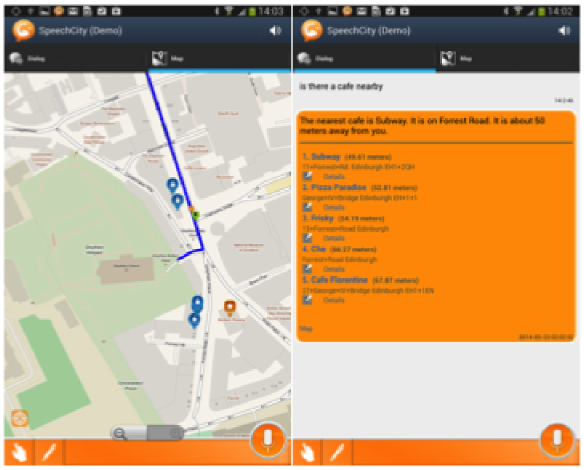

Figure 2: Multimodal SpeechCity interface.

2.2 City Modelling using Open Spatial Data

The CityModel is a spatial semantic database containing information about several hundred thousand Points of Interest in the city of Edinburgh. We harvest these entities from Open Street Maps, which are then annotated with data from social networks (FourSquare) and Wikipedia. These data include location, use class, name, street address, user ratings, number of check-ins, URL, etc. For disambiguating entities across data sources we use several factors including Levenshtein edit distance between names, geographical coordinates, address, phone number and postcode.

2.3 Multi-tasking dialogue system

The Dialogue System, consisting of an utterance parser, an interaction manager, and an utterance generator. Parsing and generation of utterances are done using hand-coded rules.

The Interaction Manager (IM) is the central component of this architecture, which provides the user with navigational instructions, pushes PoI information and manages QA questions. It handles several tasks like navigation, guided tours, weather information, amenity search and PoI information.

The IM receives the user's input in the form of a dialogue act (from the rule-based parser), the user's location (latitude and longitude) and pace rate. The location coordinates of the user are sent to the IM every 2 seconds. In addition, the IM has to deal with incoming requests and responses from the user's spoken inputs. With the possibility of system utterances being generated at a frequency of one every two seconds, there is a need for an efficient mechanism to manage the conversation and reduce the risk of overloading the user with information. In order to manage multiple conversational threads we implemented techniques such as multi-threading, prioritised queue management, and queue revision (Janarthanam and Lemon, 2014). Different dialogue threads are visually represented as separate display cards.

3 Tasks and Features

The system handles a variety of tourist information tasks such as searching for amenities (restaurants, pubs, etc), finding attractions to visit, getting directions, events, weather information, and historical information about monuments, etc. The system pushes information and recommendations according to the user's preferences and interests. For example, it might notify the user of nearby historical sights. We currently obtain user model information from the initial registration phase. In future work, we will acquire such information from the user's utterances and social network and continuously infer and update the model based on previous interactions.

4 Future Work

In future work, we aim to move away from template-based Natural Language Generation towards a domain-general framework using machine learning for generating robust instructions under uncertainty (Rieser and Lemon, 2011; Lemon et al., 2010).

Acknowledgments

The research leading to this work has received funding from the Engineering and Physical Science Research council, UK (EPSRC) under an Impact Acceleration Grant and the EPSRC project no. EP/L026775/1. For further information see `http://speechcity.com/`

References

Srinivasan Janarthanam and Oliver Lemon. 2014. Multi-threaded interaction management for dynamic spatial applications. In *Proceedings of Dialogue in Motion workshop, EACL, 2014*.

Srinivasan Janarthanam, Oliver Lemon, Xingkun Liu, Phil Bartie, William Mackaness, and Tiphaine Dalmas. 2013. A multithreaded conversational interface for pedestrian navigation and question answering. In *Proceedings of the SIGDIAL 2013 Conference*, pages 151–153, Metz, France, August. Association for Computational Linguistics.

Oliver Lemon, Srini Janarthanam, and Verena Rieser. 2010. Generation under uncertainty. In *Proceedings of INLG / Generation Challenges*.

Verena Rieser and Oliver Lemon. 2011. *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Theory and Applications of Natural Language Processing. Springer.