CityTalk: Robots that talk to tourists and can switch domains during the dialogue

Graham Wilcock

Abstract The paper describes CityTalk, a spoken dialogue system for Nao robots which provide information for tourists about local hotels, shops and restaurants. An important feature is that the robots switch domains smoothly during the dialogue if the user asks about a new domain, for example talking about hotels using a hotel database and then talking about restaurants using a restaurant database if the user asks about restaurants.
As far as we know, CityTalk is the first system to use robots with PyDial, a research toolkit for statistical dialogue systems. A topic tracker detects from user utterances when a domain switch is required. Clarification questions appropriate for different domains are based on domain-specific ontologies. CityTalk includes new domains including Tokyo restaurants and hotels.

1 Introduction

In many shopping centres in Japan, robots now greet customers and can provide information about local shops, restaurants and other facilities. Currently the Pepper robot from SoftBank Robotics is widely used. After being welcomed by the robot's polite *Irasshaimase!* greeting, customers select a topic from a touch-screen menu on the tablet computer that all Pepper robots carry around their neck. The robot then performs interaction for the selected topic, maybe giving information or playing a spoken quiz game such as Guess the Animal. When the topic is completed, users can select another topic from the touch-screen menu.

This paper describes CityTalk, a spoken dialogue system for Nao robots (also from SoftBank Robotics) that talk with tourists about local hotels, restaurants, and other places of interest. An important feature is that the robots can switch domains

---

Graham Wilcock
CDM Interact, Helsinki, Finland, e-mail: graham.wilcock@cdminteract.com

smoothly during the dialogue if the user asks about something else, without needing a touch-screen menu like the current Pepper robots.

The paper is structured as follows. After briefly reviewing related work on multi-domain systems, we describe the CityTalk robot dialogue system and the PyDial statistical dialogue toolkit which CityTalk uses. We then give example interactions, with video links and a transcript of one example, and finish with some discussion.

2 Related work on multi-domain systems

A bottleneck for dialogue systems has been the amount of data needed to manage flexible interaction, and dialogues have usually been restricted to separate closed-topic domains such as bus timetables or pizza ordering. The required knowledge is included in the system and manually structured for the purposes of the task and interaction goals. Changing to another topic is difficult, requiring either a completely new system or an extended knowledge base that includes several topics.

Many single-domain knowledge bases have been developed for various dialogue systems. However, they have been developed by different people at different locations and for different purposes, so that the knowledge representations and structures are different. The challenge is how to enable dialogue systems to deploy the existing rules or databases and smoothly switch between different topics.

Pioneering work on this challenge was done by Komatani et al. [3] using an architecture for a multi-domain system composed of several single-domain systems plus a central module. Multiple sets of rules are kept inside the distinct single-domain systems, called *domain experts*. The central module performs speech processing as well as domain selection, and it passes the speech recognition results to all the domain experts. Each expert executes its own language understanding module, updates its own dialogue states based on the given speech recognition result, and returns information required for domain selection to the central module. Based on this information, the central module selects an expert for giving the response.

For dialogue systems to be considered *open-domain* they need to be based on interactions where the user's changing interests drive the system from one topic to another. Such systems can be based on one large knowledge base containing all the possible topics that the system can talk about, such as Wikipedia, or on a large corpus of chat logs from which suitable QA pairs can be automatically learnt. An example of the first type is WikiTalk [2, 6] in which the robot talks fluently in several languages about an unlimited range of topics using information from Wikipedia. An example of the second type is a Japanese chat system by Higashinaka et al. [1] based on a large corpus of Twitter sentences. Recently intelligent speakers such as Google Home and Amazon Alexa have also combined chatting with useful activities.

3 The CityTalk robot dialogue system

CityTalk integrates the speech processing modules of the robot with the dialogue processing modules of PyDial [5], a statistical dialogue system toolkit developed by Cambridge University Engineering Department (CUED). PyDial aims to facilitate the development of statistical dialogue systems that use machine learning and deep learning techniques. The architecture of PyDial is shown in Figure 1.

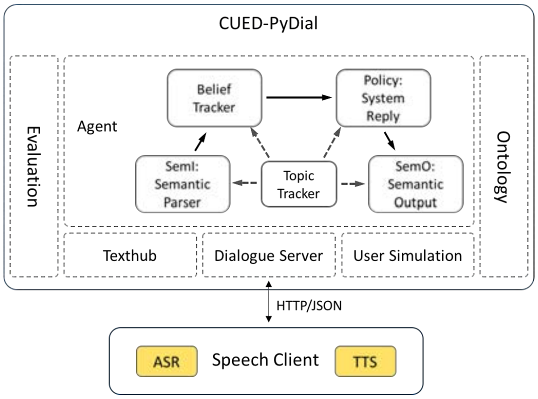

Fig. 1 The architecture of the PyDial statistical dialogue toolkit.

PyDial Agent includes a dialogue pipeline with four modules: Semantic Parser performs natural language understanding to identify the user's dialogue act, Belief Tracker updates the dialogue state, Policy module decides the dialogue act for the system's response, and Semantic Output performs natural language generation. The four modules define base classes which can have domain-specific instances.

To support multi-domain dialogues, PyDial uses a set of ontologies and databases which are domain-specific, and PyDial Agent also includes a Topic Tracker module. Instead of switching between different dialogue engines as in [3], the ontologies assist the same dialogue engine to deal with different domains. When Topic Tracker detects from the user's utterances that a new domain is required, PyDial Agent loads domain-specific instances of the classes defined by the four dialogue modules.

Speech processing is not part of PyDial, which is text-based. As Figure 1 shows, PyDial expects ASR and TTS to be performed by an external cloud-based speech client connected to its Dialogue Server module via an HTTP/JSON interface. However, CityTalk does not use PyDial's Dialogue Server with cloud speech services, because open-vocabulary speech recognition has not yet reached a satisfactory level for natural spoken interaction in real-world environments. Instead, the robot's own speech recognition and speech synthesis components are combined with techniques to increase recognition accuracy by predicting the vocabulary and limiting its size. This approach was previously successful in the WikiTalk system [2, 6].

CityTalk should detect when the user switches domains without needing to use touch-screen menus to signal the switch, and we use Nao robots which do not carry a touch-screen tablet around their neck like Pepper. When the Topic Tracker module detects a domain switch, the recognition vocabulary is modified on-the-fly for the new domain. This is possible because the speech recognition module used by Nao and Pepper robots allows the recognition vocabulary to be modified dynamically, unlike current open-vocabulary cloud speech services.

4 Example interactions

Figures 2 and 3 show example interactions with CityTalk. In the first example the robot recommends a hotel and two restaurants, using Cambridge domains from the PyDial toolkit. The user starts by asking about hotels and the robot gives information from a Cambridge hotels database. Then the user asks about restaurants and the robot switches domains and uses a Cambridge restaurants database.

Fig. 2 An example interaction with Cambridge domains, including domain-switching. The video can be viewed at https://drive.google.com/open?id=0B-D1kVqPMlKdZ3VyYUpjR3llaEU.

Fig. 3 An example interaction with the new Tokyo hotel and restaurant domains. The video can be viewed at https://drive.google.com/open?id=1qDThFaO9bOoDFO3UHfKeRlW9DaOGqGtK.

Robot:	*Hello. I can tell you about hotels and restaurants. How may I help you?*
User:	I'd like to know about restaurants.
Robot:	*What part of town do you have in mind?*
User:	In Tokyo Waterfront, please.
Robot:	*What kind of food would you like?*
User:	How about Italian?
Robot:	*Would you like something in the cheap, moderate, or expensive price range?*
User:	Ah, expensive, please.
Robot:	*Would you like something in the cheap, moderate, or expensive price range?*
User:	Are there any expensive restaurants in Tokyo Waterfront?
Robot:	*Would you like something in the cheap, moderate, or expensive price range?*
User:	Expensive!
Robot:	*I am sorry but there is no place, serving Italian food and in the expensive price range.*
User:	Oh. What about moderate?
Robot:	*I am sorry but there is no place in the expensive price range and serving Italian food.*
User:	Right... Moderate? How about moderate?
Robot:	*Cafe La Boheme is a nice place. It is in the moderate price range and it serves Italian food.*
User:	Oh, that's nice, OK. Erm, what about er...
Robot:	*Cafe La Boheme is a nice place. Their address is Aquacity Odaiba 4F, 1-9-1 Daiba, Minato-ku, Tokyo.*
User:	Oh, thank you! Thank you. Hmm, what about ...
Robot:	*Cafe La Boheme is a nice place. It serves Italian food and it is in the moderate price range.*
User:	What about hotels?
Robot:	*What part of town do you have in mind?*
User:	Again, Water..., Tokyo Waterfront.
Robot:	*Would you like the place to have parking?*
User:	I don't mind, really.
Robot:	*Would you like the place to have parking?*
User:	I don't mind.
Robot:	*Would you like a hotel in the cheap, moderate, or expensive price range?*
User:	Ah, I'd like an expensive hotel.

Table 1 Transcript of the video in Figure 3 with the new Tokyo domains.

New domains are added to the system by providing an ontology and a database for each new domain. The second example (Figure 3) uses new domains from CDM Interact (www.cdminteract.com) about Tokyo hotels and restaurants.

A full transcript of the interaction in this video is given in Table 1. The robot gives information about restaurants in Tokyo Waterfront until the user says *What about hotels*? then it switches to the Tokyo hotels domain.

5 Discussion and future work

The robot begins with the same question *What part of town do you have in mind?* for both the hotel and restaurant domains. Other robot clarification questions are domain-specific, for example *Would you like the place to have parking?* for the hotel domain, and *What kind of food would you like?* for the restaurant domain. The ontologies and domain-specific neural models or generation templates enable the robots to ask clarification questions that are appropriate for each domain.

The robot's utterances are often similar, for example when apologising because there is no suitable hotel or restaurant in the requested price ranges, but they are not simple repetitions. In each case the apology includes a list of the user's preferences, to clarify why that specific combination is not available, and the list ordering is generated separately in each case.

For scenarios where the robots will regularly be talking with tourists from many countries, it is important that they should speak as many languages as possible. This is one ability where robots will clearly go far beyond human limitations. In the Star Wars movies, C-3PO is fluent in over six million forms of communication, and in the real world Nao robots can already speak about 20 different languages. We plan to make CityTalk multilingual using methods similar to those described by [4].

Acknowledgements The PyDial statistical dialogue system toolkit was developed by Cambridge University Engineering Department. We thank the Dialogue Systems Group for making it freely available. We also thank Kristiina Jokinen of AI Research Center, AIST Tokyo Waterfront, Japan for participating in the video in Figure 3 and for valuable discussions. CityTalk was developed by CDM Interact, Helsinki, Finland (www.cdminteract.com).

References

1. Higashinaka, R., Kobayashi, N., Hirano, T., Miyazaki, C., Meguro, T., Makino, T., Matsuo, Y.: Syntactic filtering and content-based retrieval of twitter sentences for the generation of system utterances in dialogue systems. In: A. Rudnicky, A. Raux, I. Lane, T. Misu (eds.) Situated Dialog in Speech-Based Human-Computer Interaction, pp. 15–26. Springer (2016)
2. Jokinen, K., Wilcock, G.: Multimodal open-domain conversations with the Nao robot. In: J. Mariani, S. Rosset, M. Garnier-Rizet, L. Devillers (eds.) Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialogue Systems into Practice, pp. 213–224. Springer (2014)
3. Komatani, K., Kanda, N., Nakano, M., Nakadai, K., Tsujino, H., Ogata, T., Okuno, H.G.: Multi-domain spoken dialogue system with extensibility and robustness against speech recognition errors. In: Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue, SigDIAL '06, pp. 9–17 (2006)
4. Laxström, N., Wilcock, G., Jokinen, K.: Internationalisation and localisation of spoken dialogue systems. In: K. Jokinen, G. Wilcock (eds.) Dialogues with Social Robots: Enablements, Analyses, and Evaluation, pp. 207–219. Springer (2017)
5. Ultes, S., Rojas Barahona, L.M., Su, P.H., Vandyke, D., Kim, D., Casanueva, I., Budzianowski, P., Mrkšić, N., Wen, T.H., Gasic, M., Young, S.: PyDial: A Multi-domain Statistical Dialogue System Toolkit. In: Proceedings of ACL 2017, System Demonstrations, pp. 73–78. Association for Computational Linguistics, Vancouver, Canada (2017)
6. Wilcock, G.: WikiTalk: A spoken Wikipedia-based open-domain knowledge access system. In: Proceedings of the COLING 2012 Workshop on Question Answering for Complex Domains, pp. 57–69. Association for Computational Linguistics, Mumbai, India (2012)