Dialog Manager for Discover Deis: from transactions to conversations

Tuan Do, Jessica Huynh, Alex Luu, Marie Meteer
Brandeis University
415 South Street, Waltham, Massachusetts USA
{tuandn,jhuynh37,alexluu,mmeteer}@brandeis.edu

Abstract

Discover Deis is a student developed web application for navigation and location finding for the Brandeis University campus. In this paper we show how we move the application from "transactional" to "conversational" by adding a dialog manager. As the application complexity is increased, specific features in the dialog manager are required to manage that complexity, including discourse context, semantic modeling of object attributes and telic roles, and recognizing and tracking user goals. We show that the balance between application-driven and language-driven features is key to improving the dialog management aspect of a voice application such as Discover Deis.

1 Introduction

Speech application development today is very attractive because of the prevalence of mobile devices which go well with speech interfaces to provide a very natural user experience. One of the fastest ways to implement task-oriented speech applications is using web-based free resources combining both speech recognition and natural language understanding modules such as wit.ai (Dahl, 2016). This approach was applied for Discover Deis[1] (DDeis), a navigation and location finding web application for the Brandeis University campus.

Specifically, based on an interactive map, the application is capable of:

- providing detailed information about all functional places on campus (such as buildings, parking lots, athletic fields, statues, and shuttle stops), including opening hours, descriptions, and images
- detecting a user's current location as the nearest recognizable points and updating this information according to the user's movement around campus
- displaying the routes between the navigated locations with detailed navigation instruction

Currently, users can obtain the necessary information by either interacting with the menu and typing in search boxes or issuing the supported spoken commands. In either way, the interaction between the application and its users can be modeled as a series of more or less independent transactions (i.e. pairs of question and response) whose contents do not have any correlation with the previous discourse. In other words, even though we can assume certain sequential requests are from the same user, it is unnecessary for the application to keep track the history of the corresponding dialog session to carry out its available functions.

However, this simplicity, which is typical for this kind of speech application, significantly diminishes its feature richness and user-friendliness. It is clear that one of the most compelling directions to improve DDeis and other applications of its kind, especially from the perspectives of user experience and application usability, is to move from transactional to more conversational interaction between the application and its users, i.e. to integrate a more complex dialog manager (DM) into the current dialog system which is capable of truly processing discourse dynamics in a multi-dimensional manner. Following this direction, the work presented in this paper aims to prototype the most accomplishable dimensions first, which reflects the idea that application development is a process of continuous improvement with multiple iterations (Preece et al., 2015, pp. 332-333).

[1] Discover Deis was first developed as a group project in the Brandeis Summer 2015 Building Mobile Web Voice Applications program. Other developers include Tuan Do, Jessica Huynh, Alex Luu. Code implementing Discover Deis is available at https://github.com/jessicahuynh/discoverdeis.

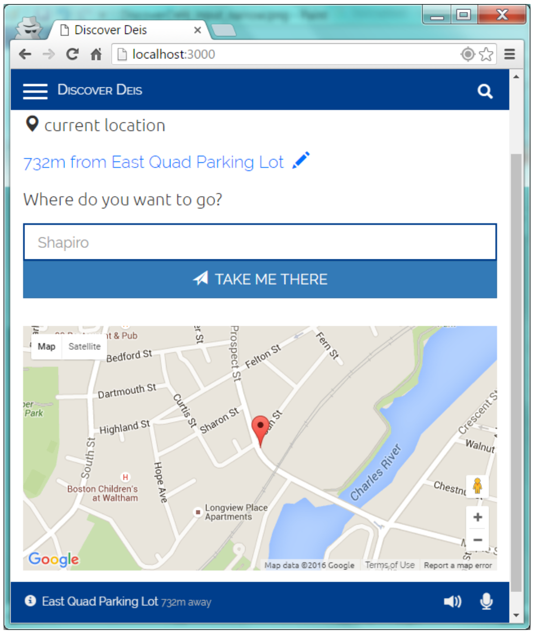

Figure 1: Transactional DDeis.

2 Dimensions of Improvement

Focusing on the conversational aspect of the interaction between DDeis and its users, we select the following dimensions of the application complexity for improvement (see Table 1).

Complexity dimension	Original DDeis	Improved DDeis
Task	Detect current location, navigate between two locations; location information inquiry	Suggest possible location of interest based on history or user role; query location for functionalities
Multi-step task	Not available	Multi-step tour, visiting multiple locations in a round, skipping/adding points
Personalization	Not available	Different types of users: visitor, student, new student, faculty
Goal drivenness	Not available	Goal managed by a ranked list of intents
Initiative	User initiative (system can ask for additional information)	Mixed initiative (system can bring up information of potential interests)

Table 1: Dimensions for improvement.

As described in Meteer et al. (2016), dimensions of complexity in an application have direct implications for the requirements of the DM. By addressing the dimensions in Table 1, the developed DM can handle the following use cases.

Dialog history (relating to the locations occurring in the history of a dialog session)

- Parking location retrieval: visitors can start their trip to Brandeis by asking for a parking lot; after that, DDeis informs them that they can get back to their parking location by saying "Where did I park?"
- Disambiguation of locations with the same name: there are some main buildings on the campus that have similar names (named after the same person). DDeis asks for clarification for first time mention, but will resolute to this first choice for next mention, with an option of reselection.

User personalization

- At the beginning of each dialog session, the system attempts to identify the user's role. The goal of this identification is twofold: first, the system will suggest potential spots of interests corresponding to the user's visiting purposes; second, certain functionalities of some buildings are limited to only certain user roles, e.g. free printing in Computer Science department may be available for students but not for visitors.

Mixed initiative

- In addition to handling a user's direct requests, DDeis can infer other implicit intents the user may have and proactively trigger these intents when they are satisfied at the user's current location. For example, when a user asks for the location of Dunkin's Donuts but then moves on to other things as Dunkin's Donuts is pretty far from her current location, intents such as "to drink coffee" and "to have a snack" are added to the user's goal list with certain ranking scores; when the user gets to a place that can satisfy at least one of these intents, DDeis will take the initiative by providing a suggestion to her such as "Would you like to have a coffee? There's a Starbucks here".

Multi-step task

- Some of the main user groups of DDeis are prospective students and their families who want to tour the campus. Locations of a tour are put in an ordered list. DDeis is capable of keeping track this multiple stop route and allowing users to add or skip any locations.

3 Additional Semantic Annotation

It is clear that to implement the additional features, we need to have a more linguistically-driven semantic representation of target entities, such as the buildings on campus. The current DDeis database is built upon Brandeis website, in which each location is accompanied by a short description and its category. It is difficult to answer these kinds of questions based on the descriptions, such as "Where can I print a poster?" or "Where can I get some coffee?" Moreover, the current poorly structured information in DDeis database is not sufficient for the design of more interactive and user-friendly conversations between the application and its users. For example, during the tour for prospective students and their families, the users could ask about a building they see in a descriptive manner, e.g. "What is the building next to the pond", or asks for verification for the current destination, e.g. "Is that the glassy one?" To address these challenges, we employ the semantic framework for nominals, proposed by Pustejovsky (1998). In this framework, a nominal is associated with a qualia structure including four factors, namely formal, constitutive, telic and agentive. For our target entities, these factors can be interpreted in terms of questions they answer:

- **Telic**: "What is the functionality of the building (given a specific user role)?"
- **Formal**: "How do you characterize the building qualitatively, e.g. using its size, color, shape and relative position?"
- **Constitutive**: "How do you characterize the building compositionally, e.g. using its materials, parts?"
- **Agentive**: "How and when was the building erected", "who was it named after?"

This semantic framework could be embedded in any level of depth. However for the simplicity of our application, we only apply it in a shallow semantic manner. For example, one could have the following semantic representation:

(1) $\text{“Farber library”}_{\text{Constitutive}} = [\text{Starbuck}]$
(2) $\text{Starbuck}_{Telic} = \{\text{anyone}: \mathit{to\ drink\ coffee}\}$

But for simplicity, we only use (1) and:

$\text{“Farber library”}_{Telic} = \{\text{anyone}: \mathit{to\ drink\ coffee}\}$

4 DDeis dialog manager

Inspired by Young et al. (2013), we model the belief state in DDeis DM in terms of the following components:

Observations (*o*)**:** This component includes intents (tasks) and entities output from wit.ai, GPS locations, and the user role as input at the beginning of a dialog session.

Intents (*i*): We currently have two categories of intents: the primary intents are those generated when users choose their role at the beginning of a dialog session, and those recognized from wit.ai through its speech API; the secondary intents are either functional intents when users asks for a place to do something, or intents inferred from primary intents by leveraging telic qualia of mentioned entities.

Goal (*g*): A ranking list of intents that get updated after each observation event. Primary intents remain in this goal list until they are satisfied or canceled (due to some explicit voiding or the conflict with newer intents). Secondary intents are characterized by their ranking scores that are decreased through time, and ultimately removed after a while. Generally, intents in the goal list are ranked according to their temporal order, i.e. the intent that needs performing first has a higher rank in the goal list.

Actions (*a*): This component is essentially the system responses. The highest-ranked intent in *g* triggers a corresponding method which, in turn, gets necessary information from DDeis database and interacts with the discourse content of the current dialog session, so-called history *h*, to perform an adequate action, which then is logged into *h* by its method parameters and outputs (i.e. what the speech synthesis produces or is shown on the screen).

History (*h*): A temporally ordered list of tuples $<o,g,a>$

This representation framework of DM demonstrates the following advantages:

- Dedicated dynamic data structure for user goal that is the key factor in DM's decision making process
- Availability of all context information of a conversation session stored in history *h* in a concise and structured manner, which makes the application of machine learning techniques possible in order to automatically learn the probabilistic model of dialog act transitions typical for this application.

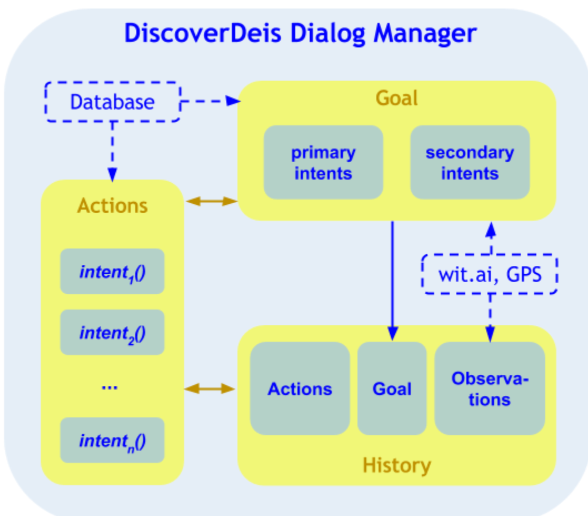

Figure 2: DDeis DM.

It is worth mentioning that the implemented prototype of current DDeis DM only focuses on domain-specific dialog control logic (cf. Bohus & Rudnicky, 2009 for two-tier dialog management framework). Specifically, all actions of DM directly correlate to particular intents (both primary and secondary) appearing in DM's goal list and are realized as intent-named methods. The overall architecture of the developed DM is presented in Figure 2.

5 Discussion

As emphasized in Meteer et al. (2016), dialog management can be driven by the application purpose or the language competence.

From the first perspective, the implemented DM is goal-driven, rather than task-driven, in the sense that it demonstrates the ability to infer a user's goal from a task request and to make decisions based on goals. As a user's goal is not always easily to prescribe in a specific hierarchical task, we handle every goal-driven unit separately, i.e. in a flat manner as shown in Actions component of DM in Figure 2 (cf. Bohus & Rudnicky, 2009 for hierarchical plan-based approach).

From the second perspective, our additional semantic annotation efficiently equips the implemented DM new capabilities such as reference resolution and name disambiguation. Overall, the balance between application-driven and language-driven features is key to improving the dialog management aspect of a voice application such as DDeis.

6 Conclusion and Future Work

In this work, we developed a working prototype for the DM of DDeis – a navigation and location finding web application for the Brandeis University campus – that succeeds in transforming the nature of the interaction between the application and its users from transactional to conversational. Specifically, this DM is capable of adding new complexity dimensions to the application, including access to dialog history, user personalization, mixed initiative, goal-drivenness and multi-step task handling. These new capabilities are achieved by the optimal enrichment of semantic representation of the available entities in the application database and the elaborate design of belief state framework of the implemented DM.

Further, we will determine the usability and acceptability of the newly developed prototype of DDeis DM by evaluating the interaction between the application and its users in terms of a variety of usability and user experience criteria. Based on the evaluation result, we will define the most relevant aspects for the next iteration of improvement.

In the long term, when a stable version of DDeis is released for public use, we will utilize the dialog content between the application and its users, including their utterances and the corresponding DM histories, as the training dataset for automatic learning of the probabilistic model of dialog act transitions typical for this application.

References

Bohus, D. and Rudnicky, A. I. 2009. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3), pp. 332-361.

Dahl, D. A. 2016. Integrating natural language resources in mobile application. In N. Shaked & U. Winter (Eds.), *Design of Multimodal Mobile Interfaces.* Walter de Gruyter GmbH & Co KG.

Preece, J., Rogers, Y. and Sharp H. 2015. *Interaction Design: Beyond Human-Computer Interaction*, 4th edn. Chichester, West Sussex, UK: Wiley-Blackwell.

Pustejovsky, J. 1998. *The Generative Lexicon*. Cambridge, Mass.: MIT Press.

Young, S., Gasic, M., Thomson, B. and Williams, J. D. 2013. POMDP-based statistical spoken dialog systems: A review. In *Proceedings of the IEEE*,101(5), pp. 1160-1179.

Meteer et al. (2016)