Dialog System for Mixed Initiative One-Turn Address Entry and Error Recovery

Rajesh Balchandran, Leonid Rachevsky, Larry Sansone, Roberto Sicconi
IBM T J Watson Research Center, Yorktown Heights, NY 10598, USA
rajeshb,lrachevs,lsansone,rsicconi@us.ibm.com

Abstract

In this demonstration we present a mixed-initiative dialog system for address recognition that lets users to specify a complete addresses in a single sentence with address components spoken in their natural sequence. Users can also specify fewer address components in several ways, based on their convenience. The system extracts specified address components, prompts for missing information, disambiguates items independently or collectively all the while guiding the user so as to obtain the desired valid address. The language modeling and dialog management techniques developed for this purpose are also briefly described. Finally, several use cases with screen shots are presented. The combined system yields very high task completion accuracy on user tests.

1 Introduction

In recent years, speech recognition has been employed for address input by voice for GPS navigation and similar applications. Users are typically directed to speak address components one at a time - first a state name, then city, street and finally the house number - typically taking four or more turns. In this demonstration we present a mixed-initiative dialog system that makes address input by voice more natural, so users can speak the complete address (in normal order) (for e.g. "Fifteen State Street Boston Massachusetts"), in a single turn. They could also specify fewer address components as per their convenience, and the system would be expected to guide them to obtain a complete and valid address.

2 System Description

Figure 1 shows the high-level architecture and key components of the system. A programmable framework consisting of a *system bus* that connects various components (called *plugins*) forms the core of the speech-dialog system. Key components include plugins to connect to the ASR (Automatic Speech Recognition) and TTS (Text-To-Speech) engines, the GUI (Graphical User Interface), the *Natural Language Processor* and the *Dialog Manager*.

2.1 Speech Recognition and component Extraction

Speech recognition is carried out using a statistical Language Model (LM) with Embedded Grammars (Gillett and Ward, 1998) to represent *Named Entities* such as city names, numbers etc. This provides flexibility for the user, while allowing for dynamic content to be updated when required, simply by swapping associated embedded grammars. For e.g., the grammar of street names could be updated based on the selected city. The IBM Embedded Via Voice (EVV) (Sicconi et al., 2009) (Beran et al., 2004) ASR engine provides this functionality and is used in this system.

In this system, a two-pass speech recognition technique (Balchandran et al., 2009) is employed, based on multiple LMs where, the first pass is used to accurately recognize some components, and the values of these components are used to dynamically update another LM which is used for the second pass to recognize remaining components. Specifically, the first LM is used to recognize the city and state while the second is used to recognize the street name and number. The street names and optionally the house number embedded grammars

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 152–155,
Queen Mary University of London, September 2009.

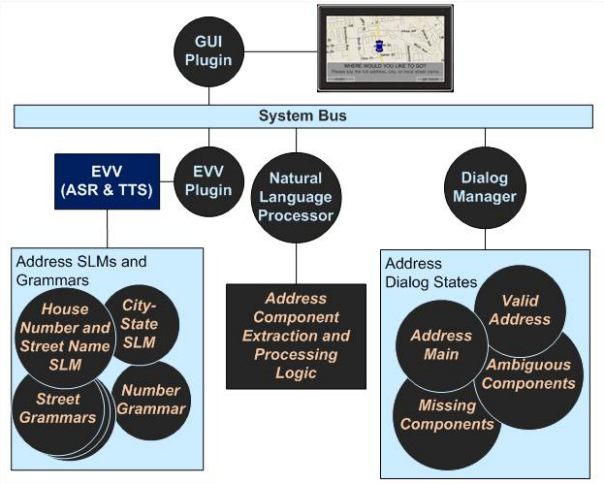


Figure 1: System Architecture

in the second LM are updated based on the city and state recognized using the first LM. This is carried out transparent to the user - so the user perceives full address recognition in one step.

2.2 Dialog management

A key part of this system is the dialog management component that handles incompletely specified input, various types of ambiguities and error conditions, all the while having an intelligent dialog with the user so as to correct these errors and obtain a valid address at the end. A goal oriented approach for managing the dialog that does not require manual identification of all possible scenarios was employed and is described in (Balchandran et al., 2009). The algorithm iteratively tries to achieve the goal (getting valid values for all address components), validating available input components, and prompting for missing input components, as defined by a priority order among components. This algorithm was implemented on a *state* based, programmable dialog manager as shown in Figure 1.

3 Scenarios

The following scenarios illustrate different situations that need to be handled by the dialog system when processing addresses.

3.1 Successful one-turn address recognition

Figure 2 shows the scenario where the user speaks a complete address in one sentence and the system recognizes it correctly.

3.2 One-turn address with error correction

The user speaks a complete address, but the system mis-recognizes the street name and number (Figure 3 (b)). The user requests to "go back" and the system re-prompts the user for the street name and number. User repeats the number in a different way (Figure 3 (c)) and the system gets it correctly.

3.3 Street and number around current location

In addition to complete addresses, the language models are built to include streets and numbers around the "current location" of the car. This data could be periodically updated based on changing car positions. In this example, (Figure 4) the user just specifies, "15 Lake View Drive" and the system defaults to the current city – Shelter Island, NY.

3.4 Ambiguous city

In this example, the user specifies an ambiguous city name (Figure 5 (a)). The system prompts the user to disambiguate by selecting a state. Once the user has done this, the system re-processes the street name and number to obtain the full address without needing the user to specify it again. The same concept is applied to other address components.

References

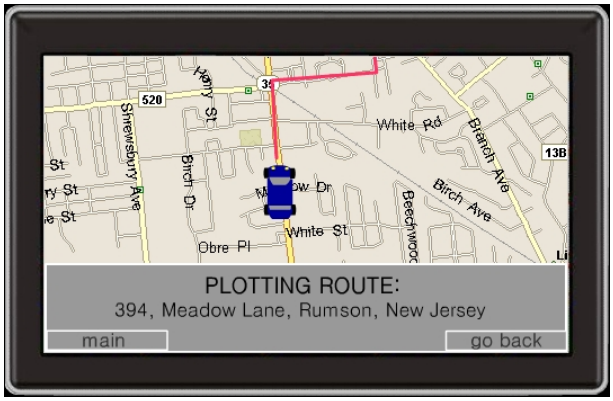
Rajesh Balchandran, Leonid Rachevsky, and Larry Sansone. 2009. Language modeling and dialog management for address recognition. In *Interspeech*.

Tomás Beran, Vladimír Bergl, Radek Hampl, Pavel Krbec, Jan Sedivý, Borivoj Tydlitát, and Josef Vopicka. 2004. Embedded viavoice. In *TSD*, pages 269–274.

John Gillett and Wayne Ward. 1998. A language model combining trigrams and stochastic context-free grammars. In *in International Conference on Spoken Language Processing*, volume 6, pages 2319–2322.

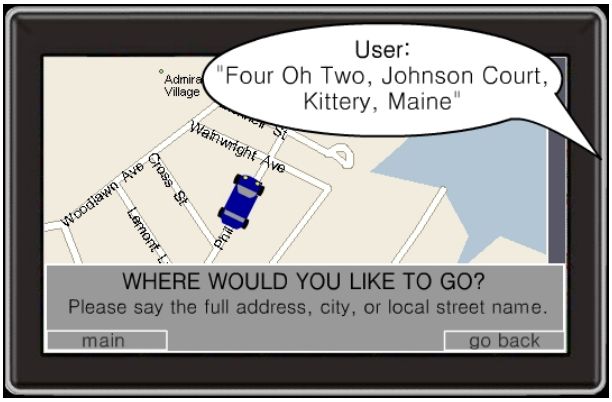
Roberto Sicconi, Kenneth White, and Harvey Ruback. 2009. Honda next generation speech user interface. In *SAE World Congress*, pages 2009–01–0518.

(a) User specifies address

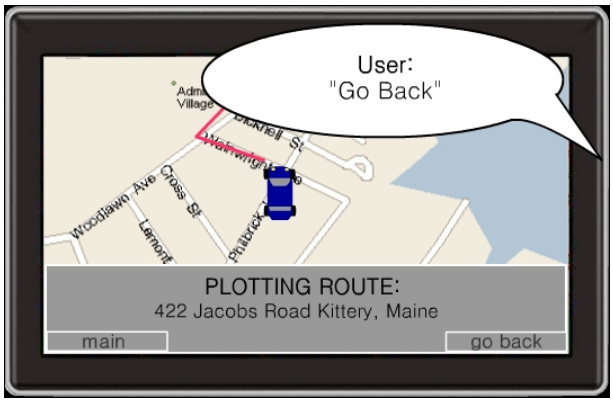


(b) System gets correct address

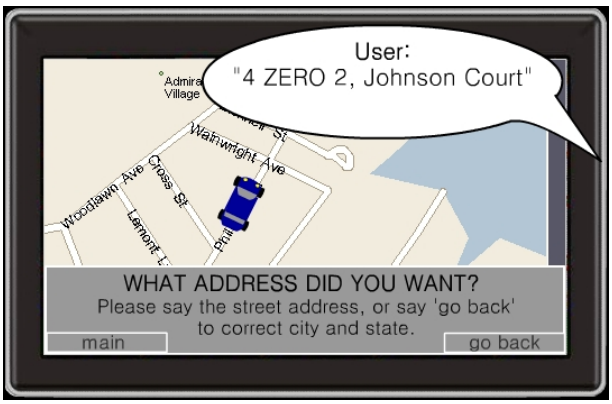
Figure 2: Successful one-turn address recognition



(a) User specifies address



(b) System makes mistake



(c) User corrects erroneous components

(d) System gets correct address

Figure 3: One-turn address recognition with error recovery

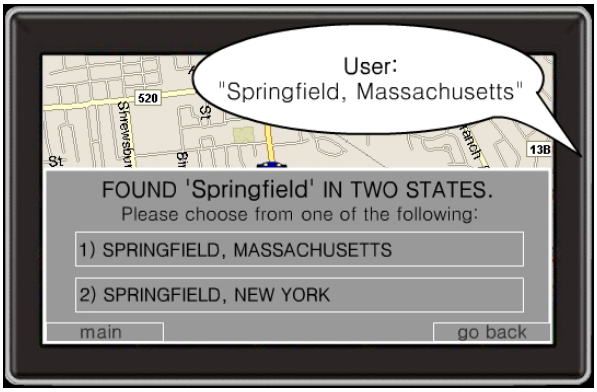
(a) User specifies street and number

(b) System locates street and number around current location

(c) System gets correct address

Figure 4: Street and number around current location (Shelter Island)

(a) User specifies address with city which is ambiguous



(b) User selects state and system combines previously specified information to get complete address

(c) System gets correct address

Figure 5: Ambiguous city example