Flexible, Robust and Dynamic Dialogue Modeling with a Speech Dialogue Interface for Controlling a Hi-Fi Audio System

Fernando Fernández-Martínez, Javier Ferreiros, Juan Manuel Lucas-Cuesta,
Julián David Echeverry, Rubén San-Segundo, Ricardo de Córdoba
Speech Technology Group, Universidad Politécnica de Madrid
E.T.S.I. de Telecomunicación, Ciudad Universitaria s/n, 28040, Madrid, Spain
{*ffm,jfl,juanmak,jdec,lapiz,cordoba*}*@die.upm.es*

***Abstract*—This work is focused on the context of speech interfaces for controlling household electronic devices. In particular, we present an example of a spoken dialogue system for controlling a Hi-Fi audio system. This system demonstrates that a more natural, flexible and robust dialogue is possible. That is due to both the Bayesian Networks based solution that we propose for dialogue modeling, and also to carefully designed contextual information handling strategies.**

***Keywords*-Spoken Dialogue System; Mixed-Initiative Dialogue Modeling; Bayesian Networks; Contextual Information.**

I. Introduction

Speech is the most widely used natural means of communication between people. Speech also is of increasing importance as a user-machine interface. As a result of the knowledge and the experience accumulated during almost half a century of research in the field of speech technology, now the time has come to design automated dialogue systems that make use of the communicative aspects of speech. In particular, it is essential to incorporate to the design of such systems some ideas related to the concept of "ambient intelligence" (AmI), for providing intelligent interfaces that are able to conduct a natural dialogue, including negotiations in order to achieve the goals required by users.

A dialogue system can be seen as a computer application that enables interaction and communication between users and machines as naturally as possible. Besides the typical recognition and text-to-speech conversion modules and other components, dialogue systems usually contain a module called Dialogue Manager (DM). This module is responsible for a dual task: to interpret the intention of the user and to decide how to continue the dialogue.

To successfully provide users with answers resembling a human-human interaction as much as possible, we believe that the design of a dialogue system should be approached from both a theoretical and practical point of view [5]. Thus, we must pay attention not only to dialogue management and modeling, but also to the enhancement of such models with knowledge about the specific tasks of the dialogue and the application domain (i.e. task and domain models). That way, it is feasible to develop procedures that support the user-machine interaction by useful elements of communication for realizing a collaborative and cooperative dialogue.

II. Prototype description

The dialogue interface that we are presenting [2], [3] allows users to drive a Hi-fi system from natural language sentences, differentially from other typical control systems based on simple commands. Thus, users can feel free to give several complex commands from a single sentence. Moreover, they don't have to memorize any command list neither use a closed specific phraseology in order to control the system successfully.

The Hi-fi audio system we are controlling is a commercial system constituted by a compact disc (with a charger of three discs), two tapes and a radio receiver. This system can be controlled by an infrared (IR) remote control. Instead, users are going to control the Hi-Fi system from a microphone. Our interface translates the speech into IR commands in order to carry out some operation or action over the system. This translation is done so that the appropriate IR commands are sent according to the user's intention.

Figure 1 shows a block diagram of our dialogue interface. The system consists of an automatic speech recognition module (ASR), which translates the audio signal into a text hypothesis of what the user has said; a language understanding module (NLU), that extracts the semantics of the user's utterance; the dialogue manager (DM), which makes use of the semantic information, together with the information gathered during previous dialogues, to determine the actions over the system that the user wants to fulfill, and to provide the user with feedback regarding the current dialogue turn; the context manager, which holds the information of the previous interactions; an execution module, that translates the actions to perform into IR commands; the response generator module (NRG), which makes use of the semantic information provided by the dialogue manager to generate a text output, and a text-to-speech module (TTS), that synthesizes the message to the user. In addition to these classical modules, we have included a speech identification module, to determine the identity of the user that is interacting with

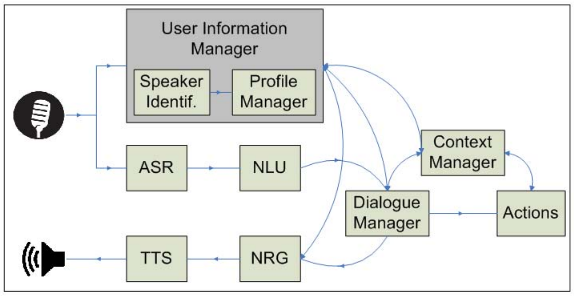

Figure 1. Block diagram of the spoken dialogue system

the system on each dialogue turn, and a manager of user-related information, that provides the dialogue manager with information related to the current user (namely, their usage permissions and preferences).

III. Demo description

A. HW resources

The hardware of the demonstration consists of a wireless microphone, a Hi-Fi system and a laptop, on which our speech interface application is installed. The laptop is equipped with a sound card and a USB remote control for sending/receiving IR commands.

B. Presentation style

The estimated duration of the demo is approximately 10 minutes. From a user's point of view, it primarily consists in addressing the system using speech in order to fulfil various dialogue goals (i.e. carry out various actions). The demo emphasizes the GUI facilities that have been specifically developed for evaluation and testing purposes. With regard to that, we will showcase, one by one, a variety of possibilities that the sytem offers for the presentation of relevant information coming from any of the modules that the system consists of.

C. Aims of the demonstration

As a result of the demonstration, we expect the audience to be aware of the following features of our system:

1) **New dialogue management approach based on Bayesian Networks** [2], [6]. As an alternative to the classical dialogue systems (finite state automata or FSMs, script based systems or dialogue plans, etc.), we are presenting a novel dialogue solution based on BNs that allow greater flexibility and naturalness by appropriately defining dialogue as the interaction with an inference system [1].
   The inference system enables **a better identification of the dialogue goals** of the user (i.e. activities or actions that the system can perform) from the available semantic information consistently with the context of the ongoing dialogue.
   In addition, BNs allow to conduct an **analysis of congruence** between the goals assumed by the system to have been requested by the user, and all data collected during the interaction. Based on this analysis **the system can determine the flow of interaction and react according to the semantics of the application domain** (e.g. performing the required tasks or asking the user for additional information if needed). The main idea is to automatically detect which concepts are needed (available or not), erroneous or optional with regard to the inferred goals. Thus the dialogue could go toward the generation of messages requesting the missing items, clarifying the erroneous ones and obviating the optional ones. This is useful for avoiding unnecessarily long dialogues and facilitates the achievement of the dialogue goals in an efficient way [4].
2) **Flexible Response**. Flexibility probably is the main asset of the proposed solution, and the most significant difference with regard to conventional approaches. In particular, **the user is not constrained to any predetermined goal or data sequence**. Thus, the BNs provide a mixed initiative dialogue modeling in which the user is free to choose at any time the goals to be accomplished by the system. This flexibility is twofold, since it not only allows the user to decide the goals at the beginning of interaction, but also lets him jump to other goals without having completed the previous ones. Moreover, the user can respond with more data than those requested in a query, or even respond to a fact not asked by the system with regard to the inferred dialogue goals. To avoid sudden changes in the interpretation (which could produce disorientation or confusion in the user) the DM must integrate all available information into the decision making process of how to continue the dialogue.
3) **Contextual Response**. Usually, systems have to deal with situations in which users omit certain information. Sometimes that information is essential for the proper outcome of the dialogue. The proposed solution allows, through **a negotiation process** based on the inference procedure, to obtain omitted information. Additionally, this solution has the ability to quickly recover the remaining information from the dialogue context. Several dialogue strategies that benefit from contextual information have been designed and implemented. That way, **the robustness of the dialogue system and the consistency of the responses with the dialogue context is improved**. These strategies are based on: the available confidence measures (both from the speech recognition and the language understanding modules), the history of the dialogue (i.e. the dialogue concepts referred so far during the dialogue), the status of the system (i.e. the current values of

the different functionalities of the system: CD, radio, volume, and so on), the task model (e.g. a semantic frame containing all the information needed to meet a specific dialogue goal), the application domain model (e.g. information on the number of tracks of a particular CD) and the user model (i.e. the information related to the current user, namely his/her preferences and privileges). Due to the designed strategies, the system is able to deal with dialogue phenomena such as "anaphora" (i.e. elements that refer to other previous parts of the dialogue) and "ellipsis"(i.e. omission of certain essential elements of the dialogue that may be derived from given context). Table I shows a possible dialog as an example of the usefulness of the history of dialogue. In particular, it is showing the possible retrieval of some resulting missing concepts. Finally, State of the system information can be helpful when no suitable entry is found in the Dialog history. A good example of this is presented in table II.

4) **Dynamic Response**. As a dynamic feature of the behavior of the system, attenuation mechanisms have been introduced that lower the relevance or the latency of information stored in past phases of the evolution of dialogue. After being stored, and due to the attenuation suffered after each dialogue turn, the relevance of these elements can evolve to a level below a predefined threshold, so that they finally disappear definitively from the dialogue history. Due to this mechanism, it is possible to **maintain the dialogue history permanently updated** by assigning higher weight to more recent information, and lower weight to older information.

   Another immediate use of this mechanism is that automatically, and without any clarification process, both erroneous and spurious elements (i.e. dialogue concepts) could be simply discarded from dialogue if these elements are no longer referenced by the user. We have included an example of a possible dialogue showing this feature in Table III.

5) **User-adaptive response**. Another feature that allows the system to response in different ways depending on the user that is interacting with it is the management of user-related information. We have included a new module, referred to as *User Information Manager* [7], which consists of a speaker identification system and a profile manager. We use the knowledge that the system has about each user to build a profile associated to that user.

   The user profile contains two types of information: *static*, such as the speaker's name, gender, age, language, and so on, and *dynamic*, used for information that changes during the dialogue. The dynamic component is composed of two different entities: *usage permissions*, which allow to add restrictions to each user (for instance, a child may not be allowed to listen to an adult-content radio tune), and *user preferences*, representing the contents that each user prefers to play (i.e. a given CD or radio tune). The user preferences keep permanently updated, with an attenuation strategy similar to that presented previously, so that the system can response to the most recent preferences of the user, giving more relevance to the last interactions.

   We have included the information of the user profiles as a new layer into the contextual information manager. Therefore, the dialogue manager will check whether there are any preference related to the current user that could be retrieved for fulfilling any goal that needs more information than the concepts that the user has provided in his/her utterance.

   An example of how the system can exploit the information stored in a user profile can be seen in table IV.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation under contracts TIN2008-06856-C05-05 (SD-TEAM UPM) and DPI2007-66846-C02-02 (ROBONAUTA).

References

[1] H.M. Meng, C.Wai and R.Pieraccini, The use of belief networks for mixed-initiative dialog modeling, IEEE Trans. on Speech and Audio Processing, 2003, vol.11, n.6, pp.757–773.

[2] F. Fernández-Martínez et al., Speech interface for controlling an Hi-fi audio system based on a bayesian belief networks approach for dialog modeling, Eurospeech, 2005, Lisboa (Portugal), pp. 3421–3424.

[3] EDECAN Project, 2006, http://www.edecan.es/en/index.html.

[4] F. Fernández-Martínez et al., Evaluation of a spoken dialogue system for controlling a Hifi audio system, Proceedings of the IEEE SLT 2008, Goa (India), pp. 137–140.

[5] F. Fernández-Martínez, PhD Thesis: Análisis, diseño y aplicación de modelos de diálogo flexibles, contextuales y dinámicos basados en Redes Bayesianas, E.T.S.I. de Telecomunicación, Universidad Politécnica de Madrid, Spain, 2008, http://oa.upm.es/1810/.

[6] F. Fernández-Martínez et al., A Bayesian Networks approach for dialog modeling: The fusion BN, Proceedings of the IEEE ICASSP 2009, Taipei (Taiwan), pp. 4789–4792.

[7] J.M. Lucas-Cuesta, et al., Managing Speaker Identity and User Profiles in a Spoken Dialogue System, Sociedad Española de Procesamiento de Lenguaje Natural (SEPLN), 43:77–84, ISSN: 1135-5948, 2009.

Table I
Concept recovery using the Dialog History.

Turn (U:user; S:system)	**Details**
U: “Play track number two”	
S: “Track number two is now playing”	
U: “Play number three”	The user omits the “track” parameter info
S: “Playing track number seven”	Unfortunately, according to the specified value, both a “track” or a “disc” are suitable parameters, however, the system is able to disambiguate the correct one between them just checking the history from more recent to older entries and retrieving the newest one
U: “Five”	Referring to the “track” parameter once again
S: “Track number five selected”	Once again the system elicits the correct parameter

Table II
Concept recovery using the State of the System.

Turn (U:user; S:system)	**Details**
U: “Load the third disc”	
S: “Loading disc number three”	From this moment on, the user is aware of what disc is selected
U: “Play track number two of the disc”	The user omits the “disc” value info
S: “Track number two is now playing”	The system checks the state of the system and retrieves the correct value, the “track” number two of the currently selected disc, assuming that the user is already aware of it

Table III
Dialog example of the attenuation procedure.

Turn (U:user; S:system)	**Details**
U: “Volume”	The user does not specify any “volume” value
S: “What do you want to do with the volume?”	The system identifies the “volume” value as “missing” and prompts the user about it
U: “Play track number five”	Actually, the user is not interested in modifying the volume
S: “Track number five is playing, would you like to do something with the volume?”	Though decreasing due to the attenuation, the remnant evidence level of the “volume” parameter is still significant enough so that the corresponding goal, e.g. “setting the volume”, is positively inferred; consequently the system continues trying to solve it
U: “Play track number seven”	New evidence decrease
S: “Track number seven playing, what would you like me to do with the volume?”	Still trying
U: “Track number nine”	After several turns the evidence level of the “volume” parameter falls below a predefined threshold, therefore the system decides to remove it from the memory
S: “Track number nine now playing”	The system stops prompting the user about the volume since only the “track selection” goal is inferred as active

Table IV
Example of dialogue session with profile manager.

Turn (U:user; S:system)	**Details**
U (John): “Switch the Hi-Fi on.”	
S: “What do you want to play, John?”	The Speaker Identifier correctly identifies John and the Profile Manager loads the information stored in his associated profile. John’s profile shows a preference of playing CD two, so it is loaded into the Context Manager, in the structure associated to the user information.
U: “Play the CD.”	
S: “Playing track one of CD two.”	As the CD two is a preference stored in John’s profile, and the system cannot recover any information from any other contextual information structure (i.e. the status of the system or the dialogue history), the system retrieves “CD 2” as feasible components to fulfill the current goal (i.e. to play a CD). Then the dialogue manager suggests and plays CD two.
U: “Now play the tape and raise the volume to five.”	John’s implicit acceptance of the new Hi-Fi settings finishes the previous dialogue. Now John wants to perform new actions. Therefore, the dialogue manager will infer two new goals.
S: “What tape do you want to play, John?”	If John’s profile does not show any preference over tape 1 over tape 2, and provided that the dialogue history does not contain any reference to “tape” (up to now, it only contains references to CD), the system has to ask John for the information it needs to perform the inferred goals (i.e. the number of tape to play).
U: “Play tape two.”	Consequently, a new dialogue turn is needed to fulfill the command that John wants to execute.
S: “Playing tape two.”	The system will annotate this new setting into John’s profile. It will constitute one of his preferences if the quotient between the number of times “tape two” has been referenced, and the number of times “tape one” has been referenced, is over a certain threshold.
S: “I’m sorry, John. You don’t have permission to set Hi-Fi volume to five. Please select another volume.”	When the system tries to solve the second goal John wants to execute, the Profile Manager retrieves that “volume 5” is a forbidden setting for him. Thus a new dialogue turn is needed in order to report John this situation, and to ask him the action he wants to perform now.
U: “Set volume to three.”	
S: “Setting volume to three.”	Finally, the system will annotate this new configuration into John’s profile.