View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Servicio de Coordinación de Bibliotecas de la Universidad Politécnica de Madrid

Facilitating Preference Revision through a Spoken Dialogue System

Juan Manuel Lucas-Cuesta[1], Asier Aztiria[2], Michael F. McTear[3], Juan Carlos Augusto[3], and Javier Ferreiros[1]

[1] Speech Technology Group, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2] University of Mondragón, Spain
[3] School of Computing and Mathematics, Faculty of Computing and Engineering, University of Ulster at Jordanstown, BT37 0QB Northern Ireland
juanmak@die.upm.es

Abstract. We present the design of a spoken dialogue system to provide feedback to users of an autonomous system which can learn different patterns associated with user actions. Our speech interface allows users to verbally refine these patterns, giving the system his/her feedback about the accuracy of the actions learnt. We focus on improving the naturalness of user interventions, using a stochastic language model and a rule-based language understanding module. The development of a state-based dialogue manager which decides how to conduct each dialogue, together with the storage of contextual information of previous dialogue turns, allows the user to speak to the system in a highly natural way.

1 Introduction

Ambient intelligence (AmI) refers to 'a digital environment that proactively, but sensibly, supports people in their daily lives' [1]. Developing systems that can interact with users in a transparent way is not only a highly attractive research topic, but also a necessity in this area.

Learning behaviour patterns is an important step towards the realization of AmI. It is essential in any non-trivial application such as domotica or healthcare. Given that any learning algorithm can produce associations which are not entirely to the satisfaction of the user, we considered it important to provide him/her a way to tune the preferences learnt by the system. The system for learning patterns of user behaviour (*PUBS*) presented in [2] was equipped with a speech interface, given that spoken language is one of the more intuitive human-computer interfaces (HCI).

That first approach only recognized simple commands to interact with the system. However, in a user-friendly HCI users want to interact in a natural way. Thus the system has to manage complex linguistic constructions, allowing the users to speak to the machine as if he/she was speaking to another human.

We propose to integrate a spoken dialogue system (SDS, [3]) into the learning system. The SDS will recognize and understand complex sentences, providing the user with a more appropriate and effective interaction tool.

2 Autonomous learning system

PUBS is able to examine a dataset of recorded events produced by sensors and actuators and produce a set of rules which describe the most frequent events and their context of occurrence. Each rule consists of three fields: an *event* that has occurred, a *condition* that should be verified, and an *action* to perform in that case.

Table 1 shows a small fragment of a dataset taken as input by *PUBS*, and a sample of a rule generated (from the whole dataset that contained the excerpt).

Table 1. Fragment of a dataset (left) and sample of a generated rule (right)

...		(ActionPattern 2)
08:02:12 Alarm	on 100	ON occurs(Bathroom, On, t_0)
08:15:54 MotionCorridor	on 100	IF context()
08:15:55 BathroomRFID	on John	THEN do(On, BathroomLights, t)
08:15:55 MotionBathroom	on 100	when $t = t_0 + 2$ sec
08:15:57 SwitchBathroomLights	on 100	(ActionPattern 3)
08:31:49 MotionBathroom	on 100	ON occurs (BathroomLights, On,t_0)
08:31:50 BathroomRFID	on John	IF context (day of week is
08:31:50 MotionCorridor	on 100	(=,(Tuesday,Thursday,Friday)))
...		THEN do (On, Shower, t) when $t > t_0$

3 User feedback system

The original speech interface, $\mathcal{I}_{PUBS}$, consisted of two open-source modules: an automatic speech recognizer (ASR), Sphinx-4 [4], and a speech synthesizer, FreeTTS [5]. $\mathcal{I}_{PUBS}$ can manage isolated and simple instructions (i.e. 'reject pattern' or 'modify condition') with which the user could refine the different fields (event, condition or action) of each learnt pattern. Although this approach achieved a first human-machine speech interaction, a more flexible SDS should benefit the system.

Our new SDS, which will be referred to as $\mathcal{I}^2_{PUBS}$, reuses the ASR and the text-to-speech (TTS) systems. To allow the user more natural utterances we have improved the language model (LM) used by the ASR. Instead of using JSGF grammars, as in $\mathcal{I}_{PUBS}$, we have collected a database of sentences of the application domain, and we have used them to train an n-gram based LM.

Besides that we have developed a natural language understanding (NLU) module, which extracts the *dialogue concepts*, that is, the semantic information of the recognized utterance, and a dialogue manager (DM), which makes use of this information to decide how the dialogue should proceed.

As we have said, the NLU is the module that parses the output of the speech recognizer and extracts the semantic information that the user has included in his/her utterance. To do that, we have developed a rule-based understanding system. This module makes use of a set of rules defined by an expert to convert a sequence of words into a set of dialogue concepts.

We have applied a two step strategy. In the first one the system labels each recognized word with the different categories that this word belongs to. We have defined a set of 20 categories, including a special one, called *GARBAGE*, which identifies words without semantic information (i.e. 'like', 'as', 'the', and so on). These words will not be considered when extracting the dialogue concepts.

Once all the words of the recognized utterance have been labeled with their corresponding categories, the second step makes use of the expert-defined rules to identify the different concepts included in the utterance. We have defined 16 different concepts associated with the application domain. Each concept is composed of an identifier and a value. Table 2 shows an example of how a given recognized utterance is labeled in both understanding steps.

Table 2. Categories and concepts extracted from an example utterance

Utterance	please set the attribute to light level
Categories	garbage, garbage, garbage, condition_value, garbage, garbage, sensor_type
Concepts	A_CONDITION:ATTRIBUTE; P_ATTRIBUTE:LIGHT

$\mathcal{I}^2_{PUBS}$ works as a finite state machine (FSM) controlled by the DM. In this way, depending on the current state and the extracted dialogue concepts, the system will take a decision about the intentions of the user (what the user has asked the system for) and the next state of the dialogue. Besides, the system will store the current information in the context database, to have it available for future dialogue turns. This information allows the system to solve different ambiguities, depending also on the state of the dialogue.

The full process can be viewed as follows. The system begins the dialogue by synthesizing a greeting. If the user requests some information (*which patterns have you learnt?*), the system returns the number of patterns and the device each pattern is associated to. The user can then ask for a single pattern (*please show me the first pattern of device A*) or for a set of them (*I want to listen to every pattern*). The system stores this information in the context database.

If the user accepts or rejects a pattern, the system updates the state of the pattern, and checks whether the user has asked for more patterns. In that case the process is repeated. Otherwise, the system ends the dialogue.

When the user desires to refine a pattern, he/she can modify any of its three fields (i.e. *could I modify the event?*, or *change the condition and the action*). He/she is then asked for the new value. After confirming the user's choice, the system checks again if any pattern remains to be presented to the user.

The user has the chance to cancel the current dialogue at any time he/she detects some wrong behaviour on the part of the system. In this case the system reports the error, acknowledges the user and ends the dialogue.

4 Discussion

We have proposed an improved speech interface to include user feedback in a learning system by means of an NLU module and an FSM-based DM. The new approach not only deals with isolated commands to refine the different patterns but also allows speakers to use more natural linguistic constructions, achieving more dynamic and user-friendly dialogues between the human and the system.

The inclusion of a context database which stores the information related to the previous dialogue acts enables the system to tackle incomplete utterances. This way the naturalness of the different dialogues is increased. Although our system is not especifically aimed at elderly people, this high degree of naturalness achieved with our approach allows anyone to interact in an easier way, so that it will especially facilitate the interaction of elderly people with the system.

Currently we are carrying out an evaluation of $\mathcal{I}^2_{PUBS}$. We are designing different scenarios in which users interact with the system performing specific actions (select a set of patterns, refine several of them and so on). We will employ different objective and subjective metrics to evaluate the performance of the system. These will include the number of dialogue turns in which a task is fulfilled as well as the recognition accuracy. Amongst the subjective metrics we are defining a questionnaire to collect the opinions of the users.

The estimation of different confidence measures for each of the extracted concepts is also a very interesting task that we want to carry out. These measures could help us in evaluating the performance of the understanding module. Besides that, the DM could use them to decide which extracted concepts are the most relevant in the current dialogue turn.

The addition of user-related information would be a highly attractive task. That is, providing the system with knowledge about the identity of the user who has caused the learning of a certain pattern. This information would allow the system to present each user with only those patterns related to him/her. Moreover, we could dynamically modify the different modules of the dialogue system to adapt their behaviour to the preferences of each user.

References

1. Augusto, J.C.: Ambient Intelligence: The Confluence of Pervasive Computing and Artificial Intelligence. In Intelligent Computer Everywhere, pp. 213–234 (2007).
2. Aztiria, A. and Augusto, J.C. and Izaguirre, A.: Autonomous Learning of User's Preferences Improved through User Feedback. In Proc. 2nd Workshop on Behaviour Monitoring and Interpretation, vol 396, pp. 72–86 (2008).
3. McTear, M.: Spoken Dialogue Technology: Toward the Conversational User Interface. Springer Verlag (2004).
4. Walker, W., and Lamere, P., and Kwok, P., and Raj, B., and Singh, R., and Gouvea, E., and Wolf, P., Woelfel, J.: Sphinx-4: A Flexible Open Source Framework for Speech Recognition. Sun Microsystems (2004).
5. Walker, W., and Lamere, P., and Kwok, P.: FreeTTS - A Performance Case Study. Sun Microsystems (2002).