Application of the ITC-irst spoken dialog system in a medical domain

I. Azzini[1], T. Giorgino[1], D. Falavigna[2], R. Gretter[2]

[1] Dept. of Computers and Systems Science
University of Pavia, 27100 Pavia, Italy
(ivano,toni)@aim.unipv.it
[2] Istituto per la Ricerca Scientifica e Tecnologica
ITC-irst, Via Sommarive 1, 38050 Povo, Trento, Italy,
(falavi,gretter)@itc.it

Abstract. The paper describes the ITC-irst approach for handling spoken dialog interactions over the telephone network. We will specifically describe the usage of the dialog system within a tele-medicine application scenario.
First, the system architecture will be summarized, then we will briefly describe our approach for evaluating confidence measures for each of the words in the "best path" provided by our recognizer. Finally, an automatic service for home monitoring of patients affected by hypertension pathology will be described. Patients must periodically introduce data into a database containing their personal medical data. The collected data are managed, according to well established medical guidelines, by an automatic system that can suggest therapies or alert doctors.

1 Introduction

The spoken dialog system developed in ITC-irst can handle mixed initiative interactions over the telephone for information access in restricted domains. In the past some prototype services were developed and tested for train timetable inquiry [1] and tourism information access [2], [3]. Since last year the system has been used for developing a tele-medicine service that will provide health assistance to chronic patients. In this system, the patients periodically call a dedicated telephone number and engage a spoken dialog with the system, which interacts with them to acquire clinical data, monitors their style of life and asks about the presence of side effects. We tried to keep the dialog as close as possible to the interaction between a physician and a patient; the service also gives advice, by issuing alerts and prompts as appropriate, to keep the patient in a low risk class. For doing this, the system is connected to a database that records the detailed medical histories of the patients being followed. We have chosen a group of patients, affected by hypertension, in care at the "San Matteo Hospital" in Pavia, Italy; they would normally have to meet their physician approximately every 2-4 weeks. The goal of the periodical examination is to monitor blood pressure values, heart rate, weight, habits and other variables used to estimate certain standard risk indicators. As suggested in [4], we have extracted the domain knowledge from a set of world-widely accepted guidelines for the hypertension and dyslipidemia pathologies. In this paper we will focus on the architecture of the dialog system and the process that we have followed for developing it.

2 The dialog system

The basic idea underlying the dialog engine lies on the definition of a set of "contexts", containing "concepts" associated to subgrammars. Therefore, a direct relation between concepts and grammars, activated during the various dialog turns, has to be established.

Our approach for language modeling makes use of recursive transition networks [5]. These are finite state networks whose arcs allow linking other grammars in a recursive way. The resulting language is context free. Since the decoding step of a speech utterance can backtrack both the grammars and the words along the best path of the language graph, the recognized string consists of a mix of words and structured information. Semantic tags are included in the recognized string as reported below:

```
Hi, my  name is (NAME(Ivano Azzini)NAME) and I have  a (PRESSURE( pressure of (VALUE(2 hundred
and 10)VALUE) )PRESSURE) and  today  I (WEIGHT(weigh (VALUE( one hundred )VALUE) kilos)WEIGHT).
```

In the string above the semantic tags NAME, PRESSURE and WEIGHT identify the related variables to collect. Note that the string above can be seen as a parse tree.

The development of the understanding part of an application basically consists in designing a set of grammars for each elementary information (concept) to collect, hence each basic concept has associated one or more grammars. In the example above, a possible grammar for concept NAME can be a list of the names and surnames linked in several ways (i.e. name-surname, surname-name, surname); for concept PRESSURE the grammar could be a list of numbers ranging within a predefined interval, and so on.

On the other hand, part of sentences that do not convey useful information (such as: "my weight is", "I weigh", "the weight today is", etc.) can be efficiently represented with stochastic language models (i.e. n-gram language models [6]). In our approach we arrange grammars into two separate levels (see figure 1). A bigram grammar, at the top level, which includes links to regular grammars, associated to the concepts, at a lower level. Initially, bigrams are trained on a small set of hand-written sentences representing (in the designer's intention) what the users will say. Once a prototype is built, field data can be recorded from user interactions. Then, the collected data are transcribed and used to update bigram probabilities. In previous experiments we demonstrated that the approach was effective for handling a domain such as access to tourism information. Refer to [2] for more details.

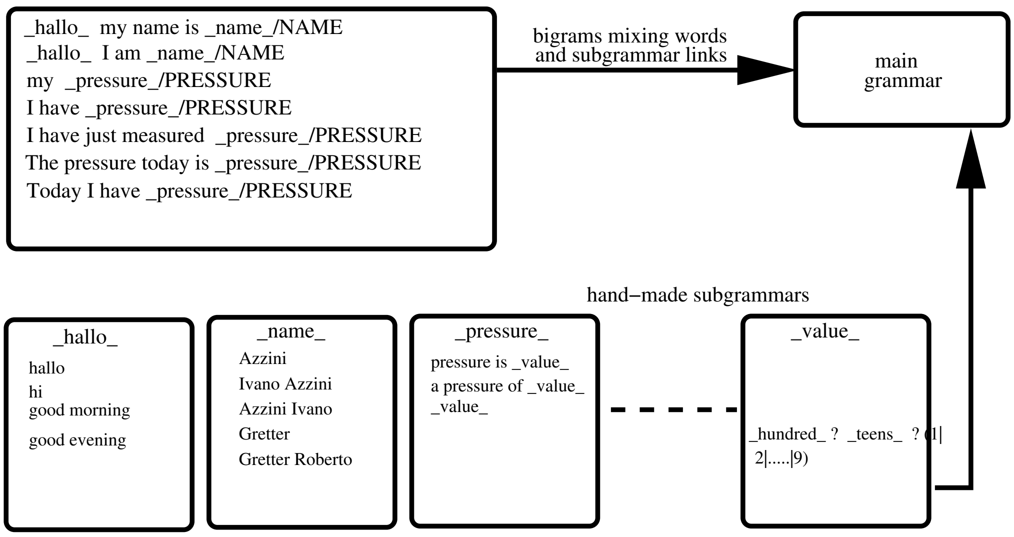

Fig. 1. Mixed language model used in the dialog system.

.

The dialog system is "frame based", i.e. basic concepts are represented by records whose fields specify a set of actions to be executed by a dialog engine. For instance, fields can contain the name of a grammar to activate for an Automatic Speech Recognition (ASR) system, the text of a voice prompt to be uttered by a Text To Speech (TTS) synthesizer, a procedure for processing the ASR output, a procedure for doing a database query, etc. In particular each concept record contains a field to be filled with the values obtained by processing the ASR output. An application is defined through a set of records and the final goal of the dialog engine is to fill some of the concepts in a consistent way. Consistency is defined through a set of boolean expressions involving both patient specific data or data collected during the interaction. Note that in this way the dialog architecture is independent of the application, since the only module to change across different domains is a file containing the description (i.e. the record concepts and some actions associated to them) of the application itself (see [2] for more details).

Presently, we are using two different languages for defining applications: the first one is proprietary, the second one is the *VoiceXML* Markup Language (see www.voicexml.org). For the tele-medicine service described below we have used the ITC-irst proprietary language.

3 Confidence measure evaluation

Currently, we are working on the introduction of confidence measures in the dialog system. We have tested three different methods for evaluating a confidence score for each word in the best string provided by the recognizer.

The first approach is based on log-likelihood ratios [7]; with this approach it is necessary to estimate an "anti-model" for each phone unit Hidden Markov Model (HMM) used in the ASR.

The second approach tries to estimate the posterior probabilities of the best path by applying the forward-backward algorithm [8] to a word lattice derived during the Viterbi decoding of the input utterance. A variation of this algorithm uses entropies instead that posterior probabilities [9].

The third approach makes use of "sausages" [10]; this is a compact representation of the word lattice, obtained by applying intra-word and inter-word clustering techniques to the lattice itself.

At present, we have introduced in our ASR confidence measures evaluated with the second approach reported above. We are presently investigating the usage of confidence scores to:

- try to avoid confirmations, if the recognized concept has a high confidence;
- try to automatically detect if the patient is going out of the scope of the dialog description;
- try to automatically update speech recognition grammars according to errors detected during the dialog progression.

We have tested all the approaches above on some some dialog databases collected in our Labs, obtaining satisfactory performance. We are going to report about it in next Conferences.

4 Patient monitoring service

Two actors are allowed to enter data into a clinical database. The physician uses a conventional (graphics, keyboard and mouse) interface to store and update data. The patient is allowed to enter data, that she/he can acquire at home, by using the telephone. To accomplish this task, the patient is instructed to dial, at fixed time intervals, a toll free number. We are currently developing a multi-modal interface, running on portable devices (such as, Personal Digital Assistant, pocket PC, etc.) with wireless connections to a Web server. This way, physicians can access the clinical database while moving inside a hospital.

One dedicated server with adequate hardware is connected to the telephone network and runs a commercial "call center" software, which answers phone calls and routes them to the ASR and TTS. When one call arrives, the dialog engine is started. After authenticating the user, it reads from a "state vector" a set of discrete variables encoding a patient specific target function. This last one determines which data must be acquired coherently with the evolution of the pathology on the basis of the knowledge related to the patient clinical history. In general, some fields of the basic concepts in the dialog description could not be filled according to patient habits (e.g. if the patient is not a smoker the system will not ask if she/he smokes, if the patient is not subjected to a pharmacological therapy the system will not ask for it, and so on). Hence, the state vector tell us which fields of the dialog description have to be acquired (mandatory) or not. Typically the target function has the following form:

Field	Value
Sport	(mandatory|optional)
Smokes	(mandatory|optional)
Blood Pressure (Dia)	(mandatory|optional)
Blood Pressure (Sys)	(mandatory|optional)
Weight	(mandatory|optional)
Heart Rate	(mandatory|optional)
Pharmacological Therapy	(mandatory|optional)
Day	(mandatory|optional)
Month	(mandatory|optional)
Time of Day	(mandatory|optional)

The target function is updated according to information contained in the database. This task is accomplished by a software agent ("adaptivity agent") connected to a relational database, which holds the detailed medical records of each patient. As previously seen, the issues considered by the agent include habits, e.g. whether she/he is a smoker; history of prescribed drugs, because one wants to check for the presence of relevant side effects; whether she is on diet, etc. Exceptional events like hang-up, multiple recognition errors and missing values ("I don't know") are also handled by the agent; this allows the relevant state information to be carried on between calls, so that the missing information can be asked the next time the patient calls, if it is appropriate.

The whole system therefore exhibits the *long-term* adaptation behavior described in [4]. This long-term adaptation for each patient takes place *after* she/he has completed the conversation with the system. The system is therefore able to adapt itself to the evolution of the disease, new patient habits or features, new physician directions, etc. We also consider a different kind of adaptation strategy, which we have called *short-term* (or punctual) adaptation. This kind of adaptation takes place *during* the course of a dialog; it can handle a range of user's behaviors, as it will be shown in the two examples below. The first example reports a piece of patient-system interaction for heart rate checking.

System: `Tell me your heart rate, please.`
Patient: `I have not measured it, but today my weight is almost 85 kilos.`
System: `So your heart rate is missing, and your weight is 85 kilos?`
Patient: `My weight is 84.`
System: `So your heart rate is missing, and your weight is 84 kilos?`
Patient: `Yes.`
System: `I would like to remind you that your physician prescribed a diet.`
`Your weight has increased by 2 kilograms since last call.`

The system, during the interaction with the patient, executes the following pseudo-code in the background.

Check *if the weight acquired is a reasonable value (e.g. not over 200 kilos):*
  False *repeat the question and try to acquire a new corrected value for the weight.*
  True *read from the patient model the information regarding "weight acquired in the last call" and "diet".*
Compute *the time elapsed since the last call and calculate a maximum allowed weight gain (loss).*
Check *if the new weight is beyond the limit evaluated above:*
  True *repeat the question and try to acquire a new corrected value for the weight.*
  False *check if the "diet" field for the patient is set to "Yes". Notify to the patient whether her/his weight has increased/unchanged/decreased since last call.*
Store *in the database the following data:*
  Weight:="84".
  Heart Rate:="Missing Data".
Go *to the next dialog step according to the target function of this call.*

The system short-term adaptation can be seen, for example, in the following case. The dialog system assumes that the calling patients are healthy; if this happens the dialog is "short", i.e. in few turns the system can obtain all the necessary medical information from the patient and the call is ended. Nevertheless, the patient could be subjected to a pharmacology therapy, which could have serious side effects. In this case the system has to manage the unexpected event “The patient is not well” by activating a more complex dialog with her/him. More specifically, after certain significant questions, a special set of grammars is activated in the background. One of them is triggered when the name of a malaise is possibly uttered. If this happens, the system reads the field “prescribed drugs” from the patient records and activates an investigation on the possible side effects for those drugs. This investigation is conducted with different levels of detail, in relation to the severity of the side effect. If the discovered malaise presents a certain gravity, the system urges the patient to immediately contact a physician .

The system has other linguistic and medical characteristics. Prompts are for example formulated in the masculine or feminine gender depending on the sex of the user; vocal prompt of different complexity are used in relation to her/his ability, etc.

5 Conclusions

Up to now, the described patient monitoring system parameters (essentially the language models) have been tuned only by means of trials carried out in the Laboratory. However, the prototype is going to be used, on the field, by a small selected set of patients subjected to hypertension. Then, we will record and transcribe all of the patient interactions and use them to evaluate performance and to update the language models and the dialog strategy.

Acknowledgment

This work has been partially done within the framework of the “Homey” project (Home Monitoring by Intelligent Dialog Systems), funded by European Union.

References

1. D. Falavigna, R. Gretter: Flexible Mixed Initiative Dialogue Over the Telephone Network. *Proc. of International Workshop on Automatic Speech Recognition and Understanding*, Keystone, Colorado, December 1999.
2. D. Falavigna, R. Gretter, M. Orlandi: Mixed Language Model for a Dialogue System over the Telephone. *Proc. of ICSLP*, Beijing, Vol. 1, pp. 585–588, October 2000.
3. C. Barbero, D. Falavigna, R. Gretter, M. Orlandi, E. Pianta: Some Improvements on the IRST Mixed Initiative Dialogue Technology. *Proc. of TSD 2000 Workshop*, Brno (Czech Republic), pp. 351–356, September 2000.
4. I. Azzini, D. Falavigna, R. Gretter, G. Lanzola, M. Orlandi: First steps toward an adaptive dialogue system in medical domain. *Proc of Eurospeech*, Aalborg (Denmark), September 2001.
5. F. Brugnara, M. Federico: Dynamic Language Models for Interactive Speech Analysis. *Proc. of Eurospeech*, pp. 1827–1830, Rhodes, Greece, 1997.
6. R. De Mori: Spoken Dialogues with Computers, Academic Press, London, 1998.
7. R. A. Sukkar, C. H. Lee: Vocabulary Independent Discriminative Utterance Verification for Nonkeyword Rejection in Subword Based Speech Recognition. *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 6, pp. 420–429, November, 1966.
8. F. Wessel, K. Macherey, R. Schluter.: Using Word Probabilities as Confidence Measures. *Proc. of ICASSP*, Seattle, USA, pp. 225–228, May 1998.
9. R. Gretter, G. Riccardi: On-line Learning of Language Models with Word Error Probability Distributions. *Proc. of ICASSP*, Salt Lake City, Utah, May 2001
10. L. Mangu, E. Brill, A. Stolcke: Finding Consensus Among Words: Lattice Based Word Error Minimization. *Proc. of Eurospeech*, Budapest, Hungary, 1999.