First steps toward an adaptive spoken dialogue system in medical domain

Ivano Azzini(+), Daniele Falavigna(*), Roberto Gretter(*), Giordano Lanzola(+), Marco Orlandi(*)

(+) Dipartimento di Informatica e Sistemistica, Università di Pavia, Pavia Italy.
(*) ITC-irst - 38050 Pantè di Povo, Trento, Italy

falavi,gretter,orlandi@irst.itc.it, ivano@aim.unipv.it, giordano@ipvaimed1.unipv.it

Abstract

Recently ITC-irst (Interactive Sensory System division) and University of Pavia (Medical Informatics Labs) are working together to realize an intelligent (adaptive) dialogue system with language understanding capabilities. In this framework, some telemedicine services able to handle multimodal interactions (i.e. input/output can be provided by both voice and/or other devices such as: keyboard, mouse, etc….) are going to be investigated and developed. Although the system is placed in medical domains, the basic concepts can be transferred towards other applications. In this papers we define the problem, explain the basic ideas, some theoretical foundation and report the early steps we have done to achieve the goal.

1. Introduction

1.1. Motivation

Speech is a natural way of communication: it is pervasive, efficient and can be remotely used; hence, it seems natural to consider the opportunity of accessing telemedicine services by telephone. Furthermore, the telephone patient - system interaction can be, in many cases, the only available type of interaction: for example if the patient is not able to use a computer, or he/she is on holiday and, in general, when computers are not available. Finally, we point out that in the next future the number of portable devices for accessing information and their corresponding process and memory capabilities will largely increase while, at the same time, their sizes (more and more small) will require the capability of handling input-output in different ways (e.g. voice input and voice or graphical output).

Nevertheless when a speech interface is available, users tend to overestimate the capabilities of a system and the users are more tempted to treat the device as another person [1]. This fact can be dramatically true in telemedicine services where the telephone access can be a direct call to the physician (or, in general, to the medical unit) or to an Automatic Speech Recognition and Understanding (ASRU) system. In this case, the patient can speak either with the physician or with the system alternatively over time: most probably, the patient will think that the system must be proficient like a physician (virtual physician!). Hence, the ASRU system - patient dialogue must be much similar to the physician - patient interaction (or, in general, human - human interaction).

To obtain this, the system must be able to show many features that we have found soundly formalized and explained in [2]. We intend to build an intelligent dialogue system, targeted at multimodal telemedicine services, with speech understanding capabilities such as to fit, most closely as possible, the McRoy’s framework. In this paper we explain the basic ideas and the first steps we have carried out to achieve the goal.

Moreover we have not found similar applications of speech recognition in medicine. The current applications of the speech technology in healthcare mainly concern systems for data input (free-text dictation in radiological reporting), data collection (endoscopic and biomedical data) and spoken interfaces for diagnostic (expert) systems (see the Pittsburgh University Division of Pathology Informatics: http://home.ncap.rr.com/voice/References.htm). Nevertheless, there is a growing interest for new applications of ASRU technology in medical domain; this work aims at a better integration between healthcare and ASRU technology through an intelligent and cooperative system.

1.2. Human - Computer Interaction Concepts

As reported in [2] the dialogue is a two-way interaction between two agents that communicate. People communicate efficiently and effectively using dialogue. Generally, people communication is driven from one or more goals (implicit or explicit, common or not common) and people tend to exchange information to each other incrementally. After each unit of communication, a speaker will pause briefly to allow listeners to provide feedback on what has just said or to add his/her own contribution to the dialogue. When given feedback, a speaker will interpret the feedback with respect to his goals and beliefs. The human - human communication is balanced and when is driven from a common goal, the dialogue time is the way of collaborating for a task or to achieve a common goal.

On the other side, human - computer communication is usually very different: computers often present large quantities of information all at a time (dialogue is not incremental) and expect relatively little feedback: in practice the system user must adapt himself/herself to the computer’s philosophy of communications!

An Intelligent Dialogue System (IDS), as defined in [2], should support the following features:

- to be able to manage an incremental dialogue between a computer system and its users;
- to monitor the effectiveness of its actions;
- to change dynamically the content to be presented as well as the system’s model of the users (for example the user’s apparent level of expertise);
- to recognize situations that require dynamic adaptation, including misunderstanding, non-understanding and argumentation.

The ITC-irst spoken dialogue system [3] allows to achieve many of the expected IDS characteristic explained above: vocal interaction is natural and intrinsically incremental; feedbacks are allowed, the flow of information between system and user is somewhat balanced, dialogue strategy and dialogue model
The system above will be placed into a multimodal telemedicine service, so as to allow many communicative forms. Nevertheless, one fundamental characteristic of an intelligent dialogue system presently lacks: “adaptation”. In order to build a full intelligent dialogue system, it is necessary to remove simple preprogrammed response actions and to obtain an incremental-cooperative human-computer dialogue.

1.3. System Adaptability

We want to investigate the meaning of adaptability inside an intelligent dialogue system for medical applications. When the system - user interaction is fixed a priori, the system is not adaptive, even if the dialogue system is incremental. Adaptability can be seen in two ways: adaptation toward different users and toward different usage situations: this consideration suggests to investigate two areas: user modeling and natural language dialogue.

The term “user modeling” denotes a model of the user maintained by the system, which correspondingly adapts its behavior. Such a system may be adaptable, if it lets the user select how the system should adapt, or self-adaptive, if it adapts autonomously, by deducing the user’s need from his/her interactions. Research on natural language dialogue is directly inspired by the thought of getting a computer to carry out a human-like dialogue. Since people are able to interact with each other in natural language, it should be natural and easy to interact with a computer in the same manner.

We can now give a general definition of an intelligent-adaptive system for medical applications: a system that is able to understand spoken language and to manage various informations, among which: the current state of the dialogue, the knowledge acquired during preceding interactions, the available information about the patient (obtained from patient model, disease model, patient record), some possible external information sources (physician directions). All those data must be used to change the system’s strategy (different strategies must be available to achieve a goal), the goal (coherently with a new and unexpected situation) and, finally, the system should be able to use all his potentiality to “force” the user to collaborate.

2. The ITC-irst spoken dialogue system

The ITC-irst spoken dialogue system [3, 4, 5] will be now briefly described. In short, it is capable of handling mixed initiative spoken dialogue interactions. The technology is based on an application-independent software architecture (see Figure 1) which essentially consists of: a telephone driver, a dialogue engine, a speaker independent continuous speech recognizer, a speech synthesizer and some minor modules. Particular care has been taken on the portability across different domains. The only modules to change across the various applications are the dialogue description (which includes the grammars used for recognition) and the database containing the information that can be required by the users. Next a brief description of the modules.

Dialogue Manager, controls the interaction flow according to a logic contained in the dialogue description file: as seen above this latter one is application dependent. The Dialogue Manager also provides a Graphical User Interface (GUI) that receives events from the various modules showing the states of the interaction and pausing or resuming the application. Telephone Driver, handles the Text To Speech engine (TTS has been developed by CSELT) and some telephone functions (e.g. on/off-hook detection, speech signal acquisition, speech signal playing, etc.) by using a commercial PC board. Automatic Speech Recognition (ASR & Spinet Server) allows to perform speaker independent continuous speech recognition using recurrent transition networks [6]. The database Manager retrieves the data required by the caller.

The recognition engine (ASR in Figure 1) is (remotely or locally) operating on a SPINET (SPEech INto Enriched Text) server and is entirely written in native code. The Spinet Server and/or recognizer generates many types of events (e.g. engine events, grammar events, result events, lexicon events, etc.); an application can handle all these events in order to build services based on automatic speech recognition. To improve flexibility and portability of ITC-irst speech technology, an Application Programming Interface (Spinet API), for requiring services (e.g. ASR, Transcriber, Grammar Compiler, etc.) to the Spinet server, has been recently developed in both Java and C++ programming languages. The Spinet API was inspired by JSAPI (Java Speech API), and defines an easy-to-use way to employ a speech recognition engine. The API consists of a set of Java classes containing methods for enabling speech recognition functions within Java applications. The Java platform offers: portability, powerful and compact environment, network awareness and security. The Dialogue Manager controls each module in the architecture and handles the communication among modules using a predefined protocol. The data contained in the description file are fed by the Dialogue Engine, by means of a Dialogue API. At present the dialogue description is written using a proprietary formalism, however an interpreter of the VoiceXML Language (the specification of this Language can be found at the url http://www.voicexml.org) for defining spoken dialogue interactions has just been developed. The approach for language modeling makes use of recurrent transition networks [6]. These are finite state networks whose arcs allow links to 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, i.e. it can be seen as a parse tree. Therefore semantic tags are included in the recognized string as reported, in the example below (a general medical data acquisition from telephone):

![Actual ITC-irst spoken dialogue architecture.](image-url)
Refined over time and is presently split into three main parts.

In Figure 2 the architecture of the intelligent dialogue system is shown, where all information and knowledge elements acquired by the multimodal system (e.g., voice, web, person to person) are stored.

Medical Unit: the physician interacting with the system. Besides the supervising task he is also responsible for tuning the dialogue system.

Electronic Patient Record: the unique long term repository where all information and knowledge elements acquired by the multimodal system (e.g., voice, web, person to person, physician interactions, etc.) are stored.

System-Patient Dialogue History: a separate storage of dialogues derived from past patient interactions carried out with the spoken dialogue subsystem. This is used for deriving statistical knowledge useful for generating new grammars, bigrams and dialogue descriptions.

Domain specific knowledge: this module is split into patient model and disease model. It encapsulates the required domain specific knowledge used to model a successful interaction of each patient with the spoken dialogue subsystem.

The SMS is the central module in the system: it is where medical domain knowledge is merged with the most technological issues of the system. SMS task is to examine all the resources in the global system in order to generate and update the medical-technological knowledge needed to obtain adaptive and cooperative dialogue descriptions.

The other system modules in the global system architecture represent the abstraction over statistical knowledge that can be achieved from the database. This knowledge is necessary for developing dialogue models capable of adapting to the specific needs and diseases of each user.

Obviously this system is able to manage traditional mixed-initiative spoken dialogue interactions for collecting data concerning a specific medical domain.

3. System Architecture

In Figure 2 the architecture of the intelligent dialogue system we are going to investigate is shown. This architecture will be refined over time and is presently split into three main parts.

On the rightmost part of the figure, the Speech Recognition, understanding and Dialogue Side (SRDS) [7] is located; the Dynamic Medical (and general) Knowledge Side’ (DMKS) is shown in the upper left while the System Manager Side (SMS) appears in the central area.

In short SMS and DMKS allow to produce dialogue descriptions for SRDS (in a proprietary language or in VoiceXML), as well as to generate new grammars or to change the database.

The basic goal of DMKS is to represent both medical knowledge and information about the temporal evolution of the interpretative analysis concerning the clinical state of each attended patient. Thus, on one side it encapsulates both the progress state reached during the conversation with the patient, as well as the results of the clinical analysis carried out in real time, as voice data are acquired and interpreted with respect to the electronic patient clinical record. On the other side however, it should also encapsulate domain specific knowledge required to accomplish the above mentioned task.

Our effort in implementing this module has been driven by the requirement of achieving an overall functional integration of all its components while preserving the logical separation among them and accounting for different formalisms for representing knowledge and information to be used in each case. This structure transpires from the arrangement of the different sub-modules, described below.

"Today my DATA (weight) DATA is about (NUMBER( 80 ) NUMBER) kilos".

In the above string the tag DATA represents the type of data provided by the user after the system’s request (weight, blood pressure, etc.), the tag NUMBER is the expected value associated with DATA. The development of the understanding part of the system basically consists in designing a set of grammars. Each basic concept has associated one or more grammars, which strictly model the concept itself. In this way the system developer has the complete control of both the concepts and the ways users are allowed to refer to them. On the other hand, hand-modeling the parts of the sentence which do not carry useful information for the task (e.g. "Today my weight is about 80 kilos" in the sentence above) is a time consuming and tedious activity. For this type of task stochastic language models are more effective. In our approach we mix the two formalisms: at the top level, a bigram grammar is activated, which is initially trained on a small set of hand-written sentences representing (in the designer’s intention) what the users will say. These sentences mix words and links to other sub-grammars [5]. The dialogue engine has to interpret the description of an application, which is both declarative (for what concerns contexts and concepts) and procedural (for the definition of the actions that must be executed in some dialogue states, for instance the preparation of a database query). Each concept has associated a set of features that specify how it will be used during the user interaction. For instance it can be associated to a label, to a grammar, possibly to a procedure for some text processing, to some vocal prompts and to some other information.

3.1. Dialogue Examples

A dialogue description is partitioned into a set of contexts (introduction, clinical data acquisition, etc.) each one including basic information, namely concepts (i.e. identification, weight, etc.) linked to grammars and voice prompts activated during the interaction. Thus an application includes any given number of basic information chunks, possibly grouped into different contexts, and the dialogue goal is to identify a coherent set of concepts. As a consequence, the system analyzes the set of concepts and builds a request message prompt for the user.

The example in Table 1 shows part of the structure for defining a dialogue [3] that has the purpose of identifying a patient.
A possible interaction that can be handled by the corresponding description is the following:

SYSTEM: Please, tell me your name.
USER: I'm Paul
SYSTEM: Good morning Paul. How are you?
SYSTEM: DB access and user validation.

Finally, possible results of this interaction could be:
- recognition error, dialogue repeats request;
- patient is correctly identified, dialogue continues over next concept;
- patient fails to be identified, dialogue stops.

3.2. An example: patients undergoing peritoneal dialysis

A small number of patients must report daily to the physician a set of clinical data: the markers of actual pathology evolution. More generally, those data are: weight, blood pressure (diastolic-systolic), cardiac frequency, replacement times, physical activity. To implement a service for these patients, a dialogue description has to be written, the service starts, acquires patient data, updates database, manages the communication (e.g. by e-mail) between patient and physician and possibly activates alarms. However, recognition grammars and voice recognizer will not be able to manage all possible interactions with users. To overcome this problem, during service usage all dialogues between users and system will be recorded in order to derive a statistical knowledge. This knowledge could enable us to adapt grammars and voice recognition parameters.

4. Future trends

Nevertheless, for many kind of applications the SMS module is useless (its only function is to link database and user trough telephone dialogues), and the global system performance is very similar to traditional telephone services such as train timetable inquiry. The insertion of some type of “adaptation” into this system is a great challenge for the new generation of telephone services. In our relatively complex system there are many different ways for effectively introducing this definition into the system. In general terms, an adaptable system should be able to understand and update automatically with scenarios changing over the time.

We are going to consider at least two different types of adaptability:
1. punctual adaptation (run-time adaptability), i.e. during a system-user dialogue, the system is able to handle user’s need and behavior never seen;
2. long-term adaptation, i.e. for each patient, after system-user dialogue, the system is able to adapt it at the disease evolution, new patient features, new physician direction, etc.

Certainly, there are many other ways to define “adaptation”, but here we have only considered a consistent definition for the medical domains we deal with.

A final remark concerns the usage of algorithms and techniques for temporal adaptation of the speech module. From this point of view the following two topics will be investigated:
1. reestimate the statistical Language Model (n-Grams), using transcribed data collected during the user interaction;
2. derive and update a statistical conceptual model of the services to deliver, using also in this case transcribed data at semantic level. Modeling the concepts of a service from a statistical point of view should provide a mechanism for improving the understanding capabilities of the overall system. At present, the IRST’s dialogue system described above does not use any statistical model of basic concepts. The statistical semantic model that we plan to use is similar to one described in [8]

5. Conclusions

This paper explains a new topic of research that involves many different research fields and open questions, so we have argued that a common theoretical problem definition was necessary. The theoretical framework, as described here, was emerged from: integration of different competences, goals identification and choices of the best strategies to achieve them. The principles explained here result from discussions between different researchers working on several areas.

6. References