Spoken dialog for e-learning supported by domain ontologies

Dario Bianchi, Monica Mordonini and Agostino Poggi
Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze 181/A
43100 Parma Italy
{bianchi,mordonini,poggi}@ce.unipr.it

Abstract

Ontology may be used in the context of e-learning to capture knowledge about learning activities and subjects. We present a system that supports an automatic speech recognition and generation between users and an agent based application on the basis these ontologies. The system is realized taking advantage of the JADE agent development framework and of the Java Speech API and has been experimented both in Italian and English.

1. Introduction

Ontologies may be used in the context of e-learning to capture knowledge about different domains.

An ontology can describe learning activities like enroll in a course, take an assessment test, join a discussion group and also the actors involved in the learning process such as students, tutors etc.

A second use of an ontology regards the description of the content of the learning material.

Many e-learning system use learning material in a standardized form of reusable learning objects [1]. These objects can be annotated by metadata to describe the content, the technology used and so on and can be stored in a repository.

The metadata used to describe the learning objects are extension of those used to describe the digital libraries, like the Dublin Core [2], and are standardized by IMS [3].

To interact whit a repository of learning material the ontology should describe both the structure of the metadata and the structure of the domain that the learning objects deal with.

With the diffusion of handheld devices and the availability on such devices of sophisticated applications that may require long interaction with the user, speech based interfaces are becoming more and more important because on the one hand, keyboards either are not available or are incompatible with a high level of portability, on the other hand, hand writing interaction is prone to errors and to tire the user.

However, speech recognition and generation is a complex task and often the interaction of the user with the application requires both a training period to teach the system to understand user's speech and additional interaction to recover errors.

In this paper, we present a system that supports an automatic generation of a spoken dialog between users and an agent based application on from the ontologies used in the application domain. Our aim is to use the ontology that is model the agent to agent communication also to model the speech interaction between humans and agents. In this case, the ontology is the source for the automatic generation of two grammars, one used for the speech recognition process and the other for speech generation. The application should be multilingual.

A number of authors reports work on the use of ontologies for some natural language processing task. Frolich and Riet [4] use different ontologies to represent the domain knowledge need in natural language generation. Wilckock [5] presented a system that generates natural language from ontologies represented in the OWL language using XSLT transformation to produce output in JMSL (Java Speech Markup Language) that can be used by a speech engine. Ontologies are also used in information extraction tasks (IE). In this case domain knowledge represented by the ontology is used to classify the entities recognized in the natural language texts and helps in the process of answer extraction [6]. Upper level ontologies are used to generate text annotation. Ontology-aware pattern matching grammars are needed for parsing texts. Finally. the use of ontologies was also proposed as a method for understand natural language conversation [7].

Ontologies are considered the basis of the communication between agents too. It was successful with the advent of speech acts based communication languages for agents [8,9].

Communication acts are also used for agent to human dialog. [10]. Different level of complexity are possible. The simple model is that of a single or system initiative dialog manager in which all the initiative is taken by the agent. In this case the context of the dialog can be represented by a finite state automata [11]. Mixed initiative dialogs arise in situations in which also the human can take the control of the dialog. They are more difficult to manage because one has to deal with belief, desire and intention models.

In many practical situation, the single initiative dialog manager is adequate. The agent can ask the human about the action to perform prompting for the required data and so on.

3. System Description

We developed a system that supports an automatic speech recognition and generation between users and an agent based application on the basis of the ontologies used in the application domain The current implementation of the system supports agent – user interaction in Italian and English, but the extension to other languages is not hard.

The system is based on the JADE agent development platform [12], the Java Speech API [13] and the IBM Via Voice tool [14]. JADE (Java Agent Development framework) is a software framework to aid the development of agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems [15]. JADE is an Open Source project, and the complete system can be downloaded from JADE Home Page [16]. The Java Speech API incorporates speech technology into user interfaces for applets and applications based on Java technology. This API specifies a cross-platform interface to support command and control recognizers, dictation systems and speech synthesizers. In particular, we have used the IBM implementation of JSAPI that provides an interface for the Via Voice tool. IBM Via Voice is a tool for the recognition and generation of speech available in many languages. Nevertheless the system does not depend on the specific JSAPI implementation.

The system is based on an agent, called SpeechManager, that allows the interaction between a user and the agents of a JADE application. To manage the conversation the SpeechManager has to maintain a context. To represent this context this agent uses a finite state automaton (implemented through a JADE FSMBehaviour). Each state maintains a “conversation act”; The possible conversation acts are:

- RecoSpellAct: speech recognition in spelling mode. The output is the recognized string.
- RecoDictAct: speech recognition in dictation mode. The output id the recognized string.
- RecoGramAct: speech recognition on the basis of a predefined grammar. The output is the tags associated to the rules of the grammar activated during the recognition.
- RecoAltsAct: speech recognition on the basis of a list of alternative strings. The output is the recognized alternative string.
- RecoIntAct: speech recognition of an integer on the basis of an integer grammar. The output is the recognized integer.
- RecoIntSpellAct: speech recognition of an integer on the basis of the integer grammar and spelling the digits. It is used for long integers and the output is the recognized integer.
- RecoFloatAct: speech recognition of a float number on the basis of a float grammar. The output is the recognized float.
- RecoDatetAct: speech recognition of a date on the basis of a date grammar. The output is the recognized date.

Specialized grammars was developed to support recognition of dates and numbers (integer and floating point both in dictation or in spelling modes). These grammars needs to be implemented for each language supported but are quite general and independent from the specific ontology.

The ontologies used by SpeechManager agents can be derived from the ones normally used by JADE agent systems in the different application domains. The ontology used by

SpeechManager consists of two kinds of elements (figure 1 shows a subset of the ontology used for a banking application):

- Concepts: entities that an agent can handle (simple or aggregate objects).
- Agent actions: the action the agents can do on the objects which are instances of defined concepts.

Both Concepts and Agent Actions have slots and filler value types.

The ontology is represented in an XML file and transformed in a class structure internally to the agent through the JADE ontology support. Instances of Concepts and Agent Actions can be created as instances of these classes. From this class structure the agent can build the dialog structure starting from an Agent Action and recursively traversing each slot (attribute) and Class to acquire the fillers (values).

At the beginning, the SpeechManager prompts the user to choose an Agent Action. Each attribute of the action is used to prompt for the values of the involved concepts. To obtain the value of a concept, the SpeechManager recursively ask the user for the value of its attributes. Recursion terminate when the requested value is a primitive type (string, number, date etc).

At the end of recognition of an Agent Action request, the SpeechManager delegates the execution of the action to an agent of the system. This delegated agent is found by querying the Directory Facilitator of the system about an agent able to perform the action.

To fill Agent Action and its attributes from speech input, it is necessary to associate with each element (Agent Action, Concepts and terminal attributes) an input modality (grammar, alternatives or spelling) and, eventually, a specific grammar. Moreover, it is also necessary to give the lexical entries (strings of words for the synthesizer) for the different languages that correspond to the names of the Agent Actions, Concepts and their attributes involved in the interaction with the user, and to recognize the different word tokens in the grammars (e.g., the different strings in alternatives).

These information are given to the agent in a form of a dictionary which should contain: a lexical entry for each action, class names and attribute names and the modality for the input of each item (dictation, spelling, list of alternatives, external grammars). Choice list and external grammars should be given for every language supported.

To avoid misunderstanding for each data entered there is a request of confirm. If a string or a number are not well recognized for two times the system automatically goes in the spelling mode. As a response to the confirmation request the user can exit the cycle and return to the initial state with the prompt for Agent Action.

SpeechManager: which action would you execute? (enroll in course, assessment test,……)
User: I want to enroll in a course.
SpeechManager: To complete the request of enroll in a course I need more information
SpeechManager: what is your first name?
User: Mark…………
…………………………….

Figure 1. Part of a possible dialog between a SpeechManager agent and a user.

4. Application to e-learning

As a part of an a e-learning system we have developed some application to give the possibility to a student to enroll in a course, to take an assessment test or to discovery the learning material he need.

In these applications the ontologies must describe the activities and actors involved in e-learning such as courses, assessment test, students, tutors and the action to perform.

Figure 1 shows a part of a possible dialog between a SpeechManager and a user.

The metadata and the domain ontology are used by the student to search the repository to find the learning objects. The student can "navigate" the repository on the basis of the concepts describing the subject of interest.

5. Conclusions

In this paper, we introduced a system that supports an automatic speech recognition and generation between users and an agent based application on the basis of the ontologies used in the application domain. The system is realized taking advantage of JADE, the leading open source framework for the development of multi-agent systems, and the Java Speech API. The system has been experimented in the field of e-lerning applications testing agent – user interaction in Italian and English, but the extension to other languages is not hard.

The method used to mapping the ontology to a dialog is very general and does not depend on the domain considered. To build a conversation agent for a new domain it is necessary only to design the ontology and to write the dictionary. Also the support to new languages is straightforward. One have only to extend the dictionary and give specialized rules to deal with numbers and dates.

A problem to address is that the dialog generated is not very flexible. To give a more usable interface it is necessary to deal with synonyms and paraphrases that the user can say.

On the other side a general purpose recognition grammar, even if useful for written text processing, is not equally valuable for speech recognition. Our approach is to generate a tailored grammar for each application domain. So we obtain a limited number or grammar rules and of lexical entries, improving the reliability of speech recognition and the robustness of the system.

References

[1] Downes S.*Learning Objects: Resources For Distance Education Worldwide*. International Review of Research in Open and Distance Learning: Vol 2 N.1 2001, http://www.irrodl.org/content/v2.1/downes.html

[2] The Dublin Core Metadata Initiative, http://www.dublincore.org/.

[3] IEEE Learning Technology Standards Committee http://ltsc.ieee.org/

[4] M. Frolich, R. van de Riet. Using multiple ontologies in a framework for natural language generation. In Proc. ECAI 98 - Workshop on Application of Ontologies and Problem Solving Methods, pp. 67-77, Brighton, UK, 1998.

[5] G. Wilcock. Talking OWLs: Toward an Ontology Verbalizer. In Proc. Workshop on Human Language Technology for the Semantic Web and Web Services, 2003, Sanibel Island, Florida.

[6] B. Popov et al. Towards Semantic Web Information Exctraction. In Proc. Workshop on Human Language Technology for the Semantic Web and Web Services. 2003, Sanibel Island, Florida.

[7] A. Sugiyama, J. Akahani, T. Satoh A Method for Understanding Natural Conversation by Using Ontology. IPSJ SIGNotes Spoken Language Processing No. 046.

[8] T. Finin, Y. Labrou. KQML as an agent communication language. In: J.M. Bradshaw (ed.), Software Agents, pp. 291-316. MIT Press, Cambridge, MA, 1997.

[9] FIPA ACL Specifications, 2000. Available from http://www.fipa.org.

[10] J. Allen. Natural Language Understanding, second edition, 1995. The Benjamin Cummings, Redwood City, CA.

[11] D. Jurafsky, J.H. Martin Speech and Language Processing, 2000. Prentice-Hall, Upper Saddle River, New Jersey.

[12] F. Bellifemine, A. Poggi, G. Rimassa. Developing multi-agent systems with a FIPA-compliant agent framework.. Software Practice and Experience, 31:103-128, 2001.

[13] Java Speech API Home Page, 2003. Available from http://java.sun.com/products/java-media/speech/.

[14] IBM Via Voice Home Page, 2003. Available from http://www-3.ibm.com/software/speech/.

[15] FIPA Specifications, 2000. Available from http://www.fipa.org.

[16] JADE Home Page. Available from http://jade.tilab.com.