Some Improvements on the IRST Mixed Initiative Dialogue Technology

C. Barbero, D. Falavigna, R. Gretter, M. Orlandi, and E. Pianta

ITC-irst, 38050 Povo, Trento, Italy,
(barbero,falavi,gretter,orlandi,pianta)@itc.it,
WWW home page: http:www.itc.it/

Abstract. The paper describes the ITC-irst approach for handling spoken dialogue interactions over the telephone network. Barge-in and utterance verification capabilities are going to be introduced into the developed software architecture. Some research activities that should enable accessing information in a new large applicative domain (i.e. the tourism domain) have been started. Objectives of the research are: language model adaptation and efficient information presentation, using a mixed representation approach. The paper reports the details of the researches.

1 Introduction

Recently, ITC-irst has developed a software architecture for information access in restricted domains [1]. The architecture is formed by some modules, namely: telephone driver, dialogue engine, automatic speech recognition, speech synthesizer, database access, graphical interface, scheduler and language generation.

Some of the functions performed by the above modules can be invoked through an Application Programming Interface (API) consisting of classes developed in either the *Java* or *C++* programming languages.

To demonstrate the effectiveness of the technology, a spoken dialogue prototype for train timetable inquiry has been developed. In this case the task consists in understanding only few concepts (i.e. departure/arrival dates and cities). Since the performance obtained on this task resulted quite satisfactory [1] a new prototype, working on a much larger domain, has been recently developed. This latter one allows to access information contained in a tourism database provided by the "Azienda Provinciale del Turismo" (APT) del Trentino, an organization that coordinates the activities of most of local operators in "Trentino Alto-Adige" (North of Italy).

2 Mixed initiative dialogue strategy

The basic idea underlying the dialogue 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 steps of the interaction, has to be established.

Our approach for language modeling makes use of recursive transition networks [2]. 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, i.e. it can be seen as a parse tree. Therefore semantic tags are included in the recognized string as reported below:

```
"I want to go in an (TYPE( hotel )TYPE) (STARS( 3 stars)STARS) in
(LOCALITY( levico )LOCALITY)".
```

In the string above the tag TYPE represents the type of accomodation requested by the caller (hotel, hostel, camping, etc.), the tag LOCALITY is the name of the place the caller wants to stay.

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. "I want to go in an" in the sentence above) is a time consuming and tedious activity. For this type of task stochastic language models are more effective.

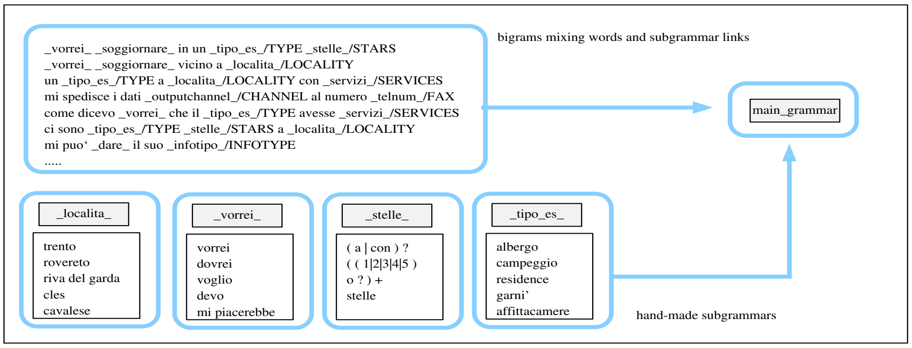

Fig. 1. *Grammars used for speech recognition.*

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 grammars (see Figure 1). As the first prototype is used to collect more and more interactions, their transcriptions can be easily added to the initial training set to build a more realistic bigram model.

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.

2.1 New system capabilities

Recently, the barge-in capability has been introduced into the prototype system. It is realized using the echo cancellation function of the telephone board: the start-end-point detection algorithm has been adapted to cope with the echo canceled signal. Some work is going to be done in order to establish some criteria for echo cancellation activation (active only during some phases of the interaction, from the beginning of the voice prompt or after a time interval from the beginning of the prompt itself, etc...).

Some preliminary work, concerning word verification, has been carried out using an approach similar to the one described in [3]. With this approach confidence measures can be evaluated for each part (e.g. a single word or sequences of words) of the recognized string. In this way the confirmation phase of the information provided by the user will be often avoided, thus rendering faster and less annoying the interaction. Some preliminary results will be reported in section 4.

Finally, to improve flexibility and portability of ITC-irst speech technology an Application Programming Interface has been recently developed for requiring services (recognition, transcription, wave source, grammar compiling, etc.) to a server. The API defines a cross-platform software interface and consists of a set of *Java classes* providing methods for enabling speech recognition functions within Java applications. The Java platform offers: portability, powerful and compact environment , network aware and security. In the same way, Java classes for handling spoken dialogue applications, according to the previously described dialogue strategy, have been added to the API. This makes easier the introduction of spoken dialogue functionalities within existing or new systems or services, especially in the ones that would be used through Internet. In particular, some financial services are going to be designed in collaboration with companies working in the field of call center products. A version of the API that uses *C++ classes* is also available.

3 Tourism information prototype

The database of APT of Trentino includes information on accomodation, structures and services, localities, events, sport, leisure time, art and culture, natural resorts, local products, holiday offers and packages, transportation. At present, we have defined and are testing a dialogue model for asking some general information about accomodation (e.g. type, category, position, services, address, etc.) and localities (e.g. altitude, description, transportation, etc.).

The user is completely free in the choice of the information to ask, both in the order the requests are made and in the linguistic expressions to use. To efficiently cope with this task the system must be capable of updating the language models, for handling the dialogue, with the large and variable data set containing the needed information. This process must be simple, allowing easy of updating also using data collected during user's interactions. Note that all the names of localities and receptive structures of Trentino are included in the system. In the future, we plan to deal with other kinds of information contained in the database (like prices of accomodation and events).

An important issue raised by the new domain is the flexibility in the presentation of the requested information. Whereas in the train timetable domain the system could give its answers using a restricted pool of fixed templates, presenting information in the tourism domain turns out to be a more complex task. In fact, tourism information is vast, rich and needs to be organized in a coherent discourse structure. Therefore, we make use of a more powerful generation component based on Mixed Representations, that is a generalization of template-based generation. This approach allows to select, for each application domain, the best trade-off between efficiency and linguistic sophistication [4].

Below some more details about the Mixed Representation approach are given. A clear distinction is generally made between deep natural language generation (based on conceptual models of language, and usually rated as flexible but inefficient) and template-based approaches (based on combination of fixed strings, efficient but not flexible), see [5]. In the last few years, templates have been used also within deep generation architectures, in order to improve efficiency ([6, 7]). All these attempts to integrate templates within deep generation architectures can be characterized as using Mixed Representations. In the adopted approach [4] we use precompiled generation knowledge whenever possible, while retaining the possibility of using a full-fledged deep generation approach when strictly necessary. For instance, let's consider the sentence *[subject] si trova a [N] metri sul livello del mare* ("[subject] is situated at [N] meters above sea level"). The Mixed Representation approach allows to restrict the use of flexible generation techniques only to the [subject] of the sentence (that could be, for example, *l'albergo a 3 stelle con sauna*, "the 3-stars hotel with sauna", or *il campeggio Bellavista*, "the camping Bellavista").

To implement the Mixed Representations approach, we use the Hyper Template Planning Language (HTPL): see [4] for a detailed description of the formalism and its interpretation mechanisms. The linguistic representation levels that can be mixed in a HTPL expression range from message representation to fixed strings. As an example, let's consider the following message: the hotel "Bellavista" has been selected during the dialogue, and the user asks for its altitude. Let's suppose the sentence *L'hotel Bellavista si trova a 1000 metri sul livello del mare* ("The Bellavista hotel is situated at 1000 meters above sea level") has to be generated. The module that plans the answer can use one of the following three templates:

1. `{message(give-information+altitude+hotel(hotel-name=X, altitude=Y))}`
2. `{message({hotel},[hotel-name=X]),‘‘si trova’’,message(attribute,`

```
{altitude=Y])}
```
3. `{''L'hotel'',X,''si trova'',''a'',Y,''metri sul livello del mare''}`

The first representation is abstract and flexible, but requires a lot of computation to be realized. On the contrary, the third representation is "ad hoc", but computationally efficient. At every time the planner can choose the most suitable representation.

4 Experiments and results

Some experiments have been carried out in order to assess the performance of the utterance verification approach. A database, formed by about 2000 speech files, has been used as a test set. Each file contains an utterance of either a name of a person or of an Italian city. All of the files have been collected through the public telephone network.

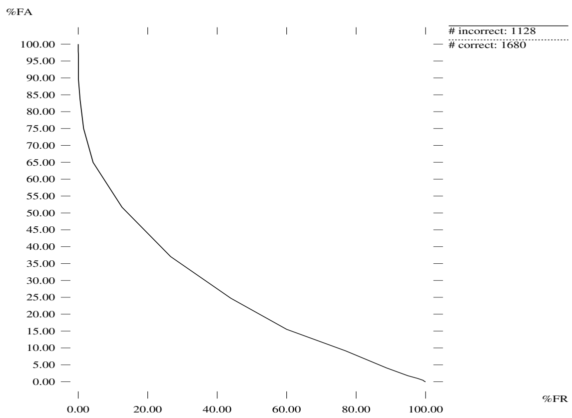

Fig. 2. *Receiving Operating Curve of the word verification test. # correct and # incorrect represent the numbers of the correctly and incorrectly recognized words respectively.*

The vocabulary of the test database consists of about 1000 words, while the grammar used for recognition is formed by a loop of about 12000 words, including the words in the test set itself. In the experiments the threshold that controls the beam width during the decoding step has been set to a value that allows to obtain a word accuracy of about 50%. In this way the number of the correctly recognized words is similar to the number of the incorrectly recognized ones. Then, confidence measures have been evaluated for each recognized word by summing the confidence scores of each hypothesized phoneme of the word itself (see [3]). To normalize the phone scores a single state Hidden Markov Model, whose output probability is represented by a mixture of 100 Gaussians, has been appropriately trained. Finally, a word verification test has been carried out by evaluating the False Rejection (%FR) rate and the False Acceptance (%FA) rate obtained with different values of the acceptance/rejection threshold (i.e. the verification threshold). The relation between %FR and %FA is shown in Figure 2.

The prototype system for accessing tourism information is presently used to collect interactions with some users. For this purpose a set of tasks has been defined: each caller is given two or three tasks. Up to now about 100 dialogues have been acquired, 34 of them have been controlled. Table 1 summarizes the state of the acquisition. First raw of the table indicates if the given task was

Table 1. *Dialogue acquisitions for accessing tourism information.*

Info obtained	(Yes) **21**	(Partially) **4**	(No) **9**			
#Turns	(1-5) **1**	(6-10) **14**	(11-15) **8**	(16-20) **2**	(21-25) **5**	(> 25) **4**
System	(1) **2**	(2) **2**	(3) **14**	(4) **8**	(5) **8**	

completely (Yes), partially or at all (No) accomplished. Note that 21, out of 34, dialogues were considered to be successful.

Second raw of the table reports the distribution of dialogues according to their corresponding number of turns. Several callers have interacted with a number of turns ranging between 6 and 10.

Third raw of the table reports the dialogue distribution according to a subjective score (1:bad, 5:good) given by the authors. The score takes into account the overall performance of the system, i.e. it is a combination of speech recognition accuracy and dialogue effectiveness.

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. F. Brugnara, M. Federico: Dynamic Language Models for Interactive Speech Analysis. *Proc. of Eurospeech*, pp. 1827–1830, Rhodes, Greece, 1997.
3. 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.
4. E. Pianta, L. Tovena: Mixing representation levels: The hybrid approach to automatic text generation. *Proc. of AISB'99 Workshop on Reference Architectures and Data Standards for NLP*, Edinburgh, pp. 8–13, April 1999.
5. E. Reiter: NLG vs Templates. *Proc. of Fifth European Workshop on Natural Language Generation*, Leiden, 1995.
6. N. Cancedda, G. Kamstrup, E. Pianta, E. Pietrosanti: SAX: Generating Hypertext from SADT Models. *Proc. of Third Workshop on Applications of Natural Language to Information Sys tems*, Vancouver, 1997.
7. S. Busemann, H. Horacek: A Flexible Shallow Approach to Text Generation. *Proc. of Nin th International Natural Language Generation Workshop*, Niagara Falls, 1998.