A MIXED LANGUAGE MODEL FOR A DIALOGUE SYSTEM OVER THE TELEPHONE

Daniele Falavigna, Roberto Gretter and Marco Orlandi

e-mail: {falavi.gretter, orlandi}@itc.it

ITC-irst – Istituto per la Ricerca Scientifica e Tecnologica
38050 Povo, Trento, Italy

ABSTRACT

This paper describes the work, under way in ITC-irst, on spoken dialogue technology over the telephone. Several activities have been carried out in order to improve either the flexibility and usage facility of ITC-irst voice recogniser and to efficiently afford difficult tasks arising when the system has to cope with large domains. At present, the dialogue technology is sufficiently robust for handling information access in restricted domains (e.g., train timetable inquiry). Furthermore, the portability of the technology towards different domains is guaranteed by an application independent architecture and by easy to use development tools and interfaces. To port the technology on larger domains (e.g., requests of tourism information) several improvements have still to be done, especially in the language models. These latter ones must be capable of handling spoken interactions aimed at retrieving information contained in large and variable databases. Hence techniques and tools for efficiently adapting language models need to be developed. Finally, language generation methods for giving the retrieved information to the callers in a useful way need to be studied.

1. INTRODUCTION

In the past [1,2] we developed a technology capable of handling mixed initiative spoken dialogue interactions. The technology is based on an application independent software architecture which essentially consists of: a telephone driver, a dialogue engine, a speaker independent continuous speech recogniser, developed in our Labs, a speech synthesiser and some minor modules. Particular care has been taken on the portability across different domains.

To demonstrate the effectiveness of the technology, a spoken dialogue prototype for train timetable inquiry was first developed [2]. In this case the task consists in understanding only few concepts: departure/arrival dates, times, cities, type of train, etc.

Since the performance obtained on this latter task resulted quite satisfactory a new prototype, working on a much larger domain, has been recently developed. This system allows to access a subset of the information contained in a tourism database, called ARIA, periodically updated with data provided by tourism operators located in “Trentino Alto Adige”, a region in the North of Italy. To cope with the new domain two problems have to be solved.

1. The language models must be capable of handling large and variable data sets – for this reason, a set of efficient and easy to use language adaptation tools must be developed, also using data collected during user interactions.

2. Information must be presented in a coherent and flexible way – i.e., a module capable of generating messages to the user starting from some structured representation of the requested information itself is needed.

This paper will only address the first of the two problems cited above: refer to [3,4] for more details about the language generation module. We are currently involved in the refinement process of the tourism domain, performing acquisitions for improving the system. However, some preliminary results will be given in section 5.

Recently, the barge-in capability has been introduced into the prototype system. It is realised using the echo cancellation function of the telephone board: the start-end-point detection algorithm has been adapted to cope with the echo cancelled 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...).

Finally, some work concerning word verification, has been carried out using an approach similar to the one described in [5]. With this approach confidence measures can be evaluated for each part (e.g., a single word or sequences of words) of the recognised string. In this way the system can avoid to ask confirmations to callers when the confidence is sufficiently high, thus rendering faster and less annoying the interaction. Some preliminary results are reported in [4].

This paper is organised as follows: section 2 describes the dialogue system architecture, section 3 deals the tourism tasks, section 4 describes the language models and section 5 reports experiments and results.

2. SYSTEM ARCHITECTURE

The architecture of our system is reported in Figure 1. The prototype can be used for information access in different domains (e.g., train timetable inquiry, tourism information inquiry, financial services, etc...), the only modules to change across the various applications are the dialogue description (which includes the grammars used for recognition) and the database. The architecture of Figure 1 consists of the following modules. Dialogue Manager, controls the spoken interaction...
flow according to a logic contained in a dialogue description file: note that only 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 pauses or resumes 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].

Database Manager, retrieves the data required by the caller. To improve flexibility and portability of ITC-ist speech technology, an Application Programming Interface (Spinet API), for requiring services (e.g. ASR, Transcriber, Grammar Compiler) to the Spinet Server, has been recently developed in the Java and C++ programming languages. In the same way, Java API for handling spoken interactions, according to the dialogue strategy, have been added (Dialogue API). 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 aware and security. The SPINET API (Speech Into Enriched Text) was inspired by JSAPI (Java Speech API), and defines an easy-to-use way to employ a speech recognition engine. The recognition engine (ASR in Figure 1) is (remotely or locally) operating on a Spinet Server and is entirely written as native code. The Spinet Server and/or recogniser generate and react in order to build services on the client's request. An application can handle all these events in order to build services based on automatic speech recognition. The system is application independent, that is only the dialogue description and the Database varies according to the application. 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 the Dialogue API. At present the dialogue description is written using a proprietary formalism, however some emerging languages (e.g. VoiceXML) are going to be introduced.

### 3. TASK DESCRIPTION

The tourism domain consists of a set of tasks designed with the purpose of accessing some of the information contained in a database, called ARIA, provided by the “Azienda Provinciale del Turismo (APT) del Trentino”, an Organisation that manages tourism activities in Trentino (North of Italy). ARIA is periodically updated with data coming from local operators. It includes information on lodging, structures and services, localities, events, sport, leisure time, art and culture, natural resorts, local products, holiday offers and packages, transportation. For this work, we have defined and tested dialogue models for asking some general information about lodging (e.g. type of an hotel, category, position, services, address, etc.) and localities (e.g. altitude, description, transportation, etc.).

During the interaction 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 hotels and camping and information about events).

Another 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. We make use of a language generation component based on Mixed Representations [4], that is a generalisation of template-based generation. This approach allows to select, for each application domain, the best trade-off between efficiency and linguistic sophistication [3].

As an example, a transcription of a part of a dialogue collected during a user’s interaction with the system is reported below.

**SYSTEM:** Good morning, automatic service for tourism information. You can ask for information on lodging; speak in a natural way.

**SYSTEM:** Hotel information: say locality, type and features of the structure you are interested on.

**USER:** I am looking for an hostel.

**SYSTEM:** I understood: type of structure hostel, do you confirm?

**USER:** Yes.

**SYSTEM:** Say the locality you are interested on.

**USER:** I want to stay in Trentino.

**SYSTEM:** There are 3 hostels in Trentino. The first is called “Benacus”, the second is called “bitta’ di Rovereto”, the third is called “giovane europa”.

**SYSTEM:** You could specify the valley or the name of the village.

**SYSTEM:** Say something.

**USER:** The village is Rovereto.

**SYSTEM:** I understood: village Rovereto, do you confirm?

**USER:** Yes, surely.

**SYSTEM:** There is an hostel in Rovereto: it is called “bitta’ di Rovereto”, it is 204 meters above sea level, its telephone number is 0464-433707.

**SYSTEM:** Say something.

**USER:** What services (that) does it offer?

…
1. DIALOGUE MODELS

Our approach for language modelling 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 recognised 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 recognised string as reported below:

"I want to go in an (TYPE: hotel) (TYPE: STARS: 3 stars) in (LOCALITY: Levico) LOCALITY*.

In the string above the tag TYPE represents the type of building 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 consists basically 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-modelling the parts of the sentence which do not carry useful information such as: "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.

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 are allowed to refer to them. On the other hand, hand-modelling the parts of the sentence which do not carry useful information such as: "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.

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 are allowed to refer to them. On the other hand, hand-modelling the parts of the sentence which do not carry useful information such as: "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.

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 are allowed to refer to them. On the other hand, hand-modelling the parts of the sentence which do not carry useful information such as: "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.

Note that the system does not impose any constrain to the caller, leaving she/he free to answer in the preferred way.

5. EXPERIMENTS AND RESULTS

The prototype system for accessing tourism information was used to collect a database of dialogues, named APT_DIAL1. To this purpose a set of about 30 tasks has been defined: each caller was given two or three tasks.

A couple of task examples are reported below.

1. You need the telephone number of the Hotel “Accademia” in Trento. Ask also for the address and try to know the number of stars of the Hotel.

2. You want to do an excursion on the “Folgaria” plateau. You want to find the mountain refuges. You need to know the telephone numbers of the refuges and the altitudes. Furthermore, you want to receive the data by Fax at the number 0461-314991.

All dialogues have been automatically transcribed and manually checked - annotations consists of word transcriptions including spontaneous speech phenomena (breaths, hesitations, truncated words) and noises of various types.

Table 1: Number of turns of the dialogues.

<table>
<thead>
<tr>
<th># dialogues</th>
<th>0-5</th>
<th>6-10</th>
<th>11-15</th>
<th>16-20</th>
<th>21-25</th>
<th>26-30</th>
<th>31-...</th>
</tr>
</thead>
<tbody>
<tr>
<td># turns</td>
<td>8</td>
<td>31</td>
<td>62</td>
<td>45</td>
<td>26</td>
<td>14</td>
<td>24</td>
</tr>
</tbody>
</table>

The total number of dialogues in the database is 230, the number of speakers is about 100, the total speech duration is 2 hours and 37 min. Complete logs of all interactions (grammars used, prompts of the system, output of the recogniser, etc..) have been retained.

The length of the dialogues, expressed in number of turns, is reported in Table 1, while the number of sentences (vertical axis) of the dialogues expressed as a function of their corresponding number of words (horizontal axis) is reported in Figure 3.

The database has been divided into a training set (3291 sentences, 11376 words) and a test set (813 sentences, 2690 words). The training set was used only for language modelling purposes; acoustic models for this experiments have been trained on our standard databases: APASCI and PHONE [1]. The baseline system has been built using hand-made subgrammars. Each basic concept is represented by one or more subgrammars. Then, bigrams have been used to model user expressions that do not carry useful information such as: “I would like”, “I am interested in finding a good place for vacation”, “I would like to stay in Trentino”, etc. During the
In a second experiment (referred as bigrams), the general grammar was built by increasing the initial 38 sentences with other 1488 training sentences (confirmations sentences were not used), obtaining a training set of 1526 sentences (6619 items). A third experiment (adapted) was settled by exploiting some knowledge of the state of the dialogue, and defining 6 bigram grammars, to be used in particular dialogue situations. Each of them was obtained by adapting the general grammar with a proper subset of sentences. In all cases, both word and semantic performances were measured. Table 2 and 3 report these data in terms of sentence and unit (words or concepts, respectively) accuracy. As expected, performance increase using bigrams.

<table>
<thead>
<tr>
<th></th>
<th>StringRR</th>
<th>UnitRR</th>
<th>Errs</th>
<th>(D+I+S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>67.40%</td>
<td>58.04%</td>
<td>420</td>
<td>(38+244+138)</td>
</tr>
<tr>
<td>bigrams</td>
<td>72.08%</td>
<td>62.24%</td>
<td>378</td>
<td>(44+233+101)</td>
</tr>
<tr>
<td>adapted</td>
<td>73.06%</td>
<td>63.44%</td>
<td>366</td>
<td>(42+222+102)</td>
</tr>
</tbody>
</table>

Table 2: Semantic accuracy.

<table>
<thead>
<tr>
<th></th>
<th>StringRR</th>
<th>UnitRR</th>
<th>Errs</th>
<th>(D+I+S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>61.25%</td>
<td>58.02%</td>
<td>1123</td>
<td>(192+209+722)</td>
</tr>
<tr>
<td>bigrams</td>
<td>66.42%</td>
<td>67.36%</td>
<td>873</td>
<td>(177+177+519)</td>
</tr>
<tr>
<td>adapted</td>
<td>67.16%</td>
<td>68.11%</td>
<td>853</td>
<td>(175+170+508)</td>
</tr>
</tbody>
</table>

Table 3: Word accuracy.

The results reported above show that:

1. the initial system (baseline) has been easily settled and has allowed to collect data at minimum cost (i.e. without involving Wizard of Oz techniques);
2. the performance of the initial system are quite satisfactory;
3. overall performance tangibly improve also using a limited number of acquisitions for estimating bigrams.

Finally, it is worth noting that with this approach it is possible to develop efficient tools that automatically adapt language models (initially defined by means of recurrent transition networks) to the given application domains.

6. CONCLUSIONS

In the paper the overall architecture of the Itc-irst dialogue system has been described. The architecture is application independent - the automatic speech recognition module and the dialogue engine module can be easily accessed through proprietary APIs. Recurrent transition networks are used to model the language in the various dialogue states. Also a mixed approach, based on both bigrams and recurrent transition network, has been introduced. The approach allows to significantly improve performance with respect to the baseline system. Tools for language model creation and adaptation, starting from the transcriptions of real interactions have been developed and are going to be refined. The introduction of either barge-in capability and confidence measures is expected to augment the naturalness of the interaction.

7. REFERENCES