Development and evaluation of the DIHANA project dialog system

D. Griol, F. Torres, L. Hurtado, S. Grau, E. Sanchis, E. Segarra

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València, E-46022 València, Spain
{dgriol,ftgoterr,lhurtado,sgrau,esanchis,esegarra}@dsic.upv.es

Abstract

In this paper, we present our work in order to develope and evaluate a dialog system in the DIHANA project. This system consists of seven modules: an automatic speech recognizer, a language understanding module, a dialog manager, a database query manager, a natural language answer generator, a text-to-speech converter and, finally, a central communication manager.

We review the current characteristics of the modules that compose our dialog system and the methodologies and strategies developed to carry out its specific functionality. Finally, our work in order to evaluate the operation of the dialog system is presented.

1. Introduction

In the last few decades, the development of speech technologies has led to speech-based solutions for several tasks. Although the current state of speech technologies does not allow the construction of general dialog systems, domain restricted dialog systems have been becoming feasible in the last decade.

The scheme used for the development of these systems usually includes several generic modules that deal with multiple knowledge sources and that must cooperate to satisfy user requirements.

The main goal of the DIHANA project [1] is the development of a robust, distributed and modular dialog system for access to information systems. The task of this project is to provide information in natural language about train services, schedules, and fares in Spanish.

In this paper, we present the basic characteristics and methodologies of the modules of the dialog system developed for the DIHANA project, and the techniques and measures used to evaluate the system operation.

2. The acquisition of the DIHANA corpus

In order to learn probabilistic models for the main modules in the dialog system there must be an annotated corpus of the task available. This implies the acquisition of a dialog corpus of the task and the labelling of the dialogs. The acquisition of the DIHANA corpus was carried out using the Wizard of Oz technique (WoZ).

This acquisition was not restricted at the lexical and syntactical level (spontaneous speech), but at the semantical level (i.e., the acquired dialogs are related to a specific task). In our acquisition, this semantic control was also provided by the definition of scenarios that the user must accomplish and by the WoZ strategy, which defines the behaviour of the acquisition system and that is modeled by one of the dialog managers presented in this paper.

A platform was constructed for the acquisition of dialogs. This distributed and modular platform was available from the BASURDE project [2], where all the modules are completely automatic. In our case, the dialog manager module was substituted by the Wizard, which used the acquisition tool.

For the acquisition process, 225 volunteers were recruited, each of them acquiring four scenarios (which were balanced to have different types and both open and closed variants). Most of the acquirers were external to DIHANA and ignored the existence of the Wizard.

The acquisition process resulted in a spontaneous Spanish speech dialog corpus with 225 different speakers (153 male and 72 female), with small dialectal variants. On average, each dialog consisted of seven user turns and ten system turns, with an average of 7.7 words per user turn. The vocabulary size was 823 words. The total amount of speech signal was about five and a half hours. This corpus has been recently used in order to develope new stochastic strategies for the dialog management, as well as for the design of the models of the main modules in the DIHANA dialog system.

Each volunteer was requested to give their opinion about the system operation and their sensations after using it. The data collection was made using surveys that the users complete after their interaction with the system. Some of the questions included in this survey were the following:

- Did you understand the system when it spoke?
- Did the system understand what you said?
- Was the interaction rate appropiate?
- Did you know what to do in every moment of the dialog?
- How often the system was slow given the answer?
- Did the system operate as you expected during the conversation?
- Was it easy to obtain the objective of the different scenarios?

This information was used to improve the adquisition process and the operation of the different modules of the adquisition platform by taking into account the users opinion and their sensations about the system operation. Therefore, the corpus that was adquired implicitly incorporates these opinions.

3. The DIHANA dialog system

3.1. System Architecture and communication data packages

Figure 1 shows the system architecture. In this figure, we use the following acronyms: ASR (Automatic Speech Recognition module), SU (Speech Understanding module), DM (Dialog Manager), DQM (Database Query Manager), AG (Answer Generator module), and TTS (Text-To-Speech synthesizer).

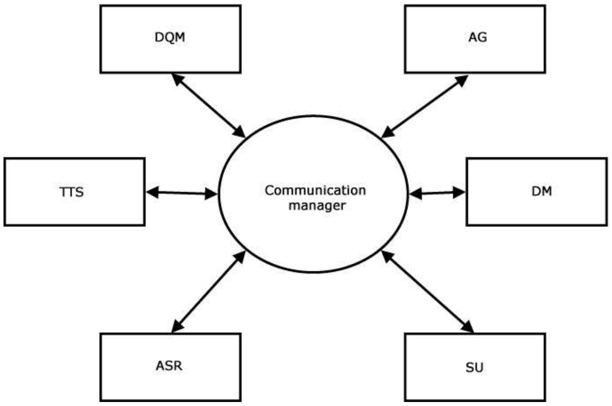

Figure 1: Description of the system architecture.

Communication among the modules is done by means of sending XML communication data packages through sockets. The information about the origin module and the destination module are in the header of these data packages.

Figure 2 shows an XML communication data package that is used in the dialog system. This package comes from the automatic speech recognition module (ASR) and goes to the speech understanding module (SU).

```
<?xml version=“1.0“ encoding=“ISO-8859-15“?>
<package destination=“SU” origin=“ASR” version=“1.0”>
  <recognized_sentence>
    <information>
      I would like the train fares from Valencia to Madrid
    </information>
    <confidences>
        <word confidence=“0.47” value=“I” />
        <word confidence=“0.68” value=“would” />
        <word confidence=“0.53” value=“like” />
        <word confidence=“0.75” value=“the” />
        <word confidence=“0.64” value=“train” />
        <word confidence=“0.56” value=“fares” />
        <word confidence=“0.84” value=“from” />
        <word confidence=“0.93” value=“Valencia” />
        <word confidence=“0.78” value=“to” />
        <word confidence=“0.93” value=“Madrid” />
    </confidences>
  </recognized_sentence>
  <grammar name=“dihana.jsgf”>
</package>
```

Figure 2: Example of XML data packages.

3.2. Automatic Speech Recognition (ASR)

The automatic speech recognition module (ASR) transforms the user utterance into the most probable sequence of words. We used the Sphinx utilities [3] from Carnegie Mellon University to train the acoustic models and decode the user utterance.

We trained the acoustic models using the SphinxTrain program. We used semicontinuous acoustic models using 25 phonemes plus silence for the Castilian-Spanish. The acoustic training was performed using the 4 hours, 47 minutes of spontaneous speech utterances of the DIHANA corpus.

We also used a Witten-Bell discounted trigram language model constructed using the CMU-Cambridge Language modeling tookit [4]. We used the 6,280 user utterances from the DIHANA corpus. The total number of different words was 812. We applied a categorization process with natural categories like CITY-NAME, TRAIN-TYPE, DAY, MONTH, etc. Then, the vocabulary size was reduced to 649 different words.

We used Sphinx-II as the decoder. The word error rate obtained was 14.07%.

3.3. Speech understanding (SU)

As in many other dialog systems, the semantic representation chosen for the task is based on the concept of frame. Therefore, the understanding module takes the sentence supplied by the recognition process as input and generates one or more frames, with the corresponding attributes, as output. In this task, we identified 11 concepts. Some of them are: DEPARTURE-HOUR, ARRIVAL-HOUR, FARE, etc. Each concept has a set of attributes associated to it (ORIGIN, DESTINATION, DEPARTURE-HOUR, ARRIVAL-HOUR, TRAIN-TYPE, etc). This set represents the restrictions that the user can place on each concept in an utterance.

The speech understanding module (SU) is based on stochastic models estimated by means of automatic learning techniques [5]. The understanding process is done in two phases:

1. The first phase translates the input sentence into a semantic sequence defined in an intermediate language (ISL), using a stochastic model.
2. The second phase translates this semantic sequence into its corresponding frame (semantic representation, concepts and attributes, used to commmunicate with the dialog manager). This process is based on the use of rules that guarantee the order of the translation to frames.

Figure 3 shows the result of applying the two phases of the understanding module to the user utterance *“I would like the train fares from Valencia to Madrid”*.

```
(FARE)
  ORIGIN: Valencia
  DESTINATION: Madrid
```

Figure 3: An example of translation into concepts and attributes.

3.4. Dialog manager (DM)

Different strategies have been designed to develope the dialog manager module (DM).

The first strategy is based on the use of stochastic models. The states and transitions in the model are learned from the BASURDE corpus [2], which is a corpus that refers to the same task. The selection of transitions in the model are determined by the user information given in the corresponding turns, on the basis of the semantic representation generated by the understanding module. This dialog manager determines the system strategy using two components, a stochastic dialog model (SDM) and a historic register (HR). The scarceness of the training corpus causes the SDM to have a partial knowledge of the task event space. In consequence, the dialog manager follows a hybrid strategy, which is partially stochastic corpus-based, and partially fixed by a set of rules. A detailed explanation of this dialog manager can be found in [6].

In our project, we also considered the development of rule-based dialog models, which are portable and adaptable to other tasks. The practical implementation of these models is done by using dictionaries and files for the definition of the semantics of the task and in the determination of a standard format for the output of the dialog manager and for the communication among the modules in the system. Different models have been developed based on the strategy followed to confirm the values provided by the user whenever the manager considers that their reliability is not high enough.

One of these rule-based dialog managers models the strategy that was defined for the acquisition of a data corpus with the Wizard of Oz technique, in which a human imitates the behavior of the system. This strategy is based on the confirmation of the values provided by the user whenever the manager considers that its reliability is not high enough. A detailed explanation of this dialog manager can be found in [7].

Both methodologies use confidence measures, which are provided by the understanding module, to determine the data reliability [8]. The use of confidence measures for error correction in the dialog management has also been proposed by other authors [9], [10].

3.5. Answer generator (AG)

The answer generator module (AG) translates the semantic representations of the system turns to sentences in Spanish. It uses templates and combines rules to make this translation.

The input of the answer generator is composed of concepts and attributes (as in the understanding module) with confidence measures associated to each frame. These measures allow the generation of detailed answers in natural language.

The sentence in natural language generated by the AG module is sent to the TTS module, which makes the text conversion to voice in two phases. The first phase analyzes the input text to generate its phonetic transcription, including additional information about duration, intonation, and rhythm. The second phase processes the information that is received from the first phase and generates the suitable signals. The final waveform is obtained by means of the concatenation of voice segments previously recorded in the form of diphonemes. Finally, the waveform is modified to adapt it to the prosody of the text.

3.6. Database query manager (DQM)

A database that follows the relational data model has been designed using the PostGreSQL database manager. The design of this database was made taking into account the specific information requirements of the DIHANA project, as well as the solutions proposed by other real systems for the same task. In the designed database, the information is structured in 11 different tables that contain information about stations, train types, ticket types, train routes, ticket fares, and user services, as well as the interrelations among all these elements.

In this database, those trips that have the same origins, destinations, schedules, and prices, are grouped together regardless of their date. This design allows us to have information about more than 400,000 independent trips in a manageable database.

The database query manager (DQM) receives a request for information from the dialog manager as input and gives back an information structure that represents the information that has been required as a result. This module constructs the query. Once the query is executed, it interprets and structures the results.

3.7. Central communication manager

This module is responsible for contacting the rest of the modules in the system. It receives all the messages sent and directs them to the destination server. It acts as a guide for the messages that are transmitted by the different modules and is responsible for establishing the communications and showing the information generated in the dialog.

In order to show this information to the user, two information blocks have been distinguished taking into account their contents. The first block shows the control information that reflects the state of the different modules that make up the system and all the transmitted messages. The second block informs the user about the sentences that have been recognized and the answers generated by the system after each intervention.

4. System evaluation and future work

The evaluation of the behavior of our dialog system was made using a set of scenarios consisting of different queries about timetables and/or prices of one-way trips. We considered the following measures:

1. Dialog success rate (% success). This is the percentage of successfully completed tasks. In each scenario, the user has to obtain one or several items of information and, then, the dialog success depends on whether the system provides correct data (according to the aims of the scenario) or incorrect data to the user.
2. Average number of turns (nT). This is the average of system turns per dialog.
3. Confirmation rate (% confirm). This was obtained by counting the explicit confirmation turns, nCT, per dialog system turn, that is, nCT/nT.
4. Average number of corrected errors per dialog (nCE). This is the average of errors detected and corrected by the dialog manager. We have counted only those errors that modify the values of the attributes (and that could cause the failure of the dialog).

The different dialog managers strategies were evaluated and the values of the previous measures were obtained. We have currently made an evaluation of the system using a set of expert users, who knew the main characteristics of the task and the basic operation of a spoken dialog system. In this context, the results are quite satisfactory.

A set of 150 dialogs (15 dialogs for each scenario) were performed by expert users to evaluate the dialog system using the stochastic dialog manager that we have presented. High success rates (99% and 69%) were achieved , even when the received user frames had a significant error rate (20% and 30%, respectively).

The rule-based dialog manager was tested with 120 dialogs developed by six expert users. The understanding module provided inputs to the dialog manager with a word accuracy of 80%. The results of this evaluation show the satisfactory operation of the dialog manager in this context, with a success rate of 93%.

Once we have the complete prototype, the next step will be to evaluate the system with real users. The only information that we currently have related to users not familiarized with this kind of systems was obtained during the adquisition of the DIHANA corpus.

Once a complete system is available, new works are currently being developed in a new project called EDECAN [11]. The goals of these works are the following: to improve the system architecture, to make the integration of the automatic speech recognition and speech understanding tasks in only one module, to improve the graphical interface of the system and to adapt the different modules to carry out different tasks.

The EDECAN project is aimed at exploring techniques to increase the robustness of a spontaneous speech dialog system. EDECAN targets the development of technologies for adaptation and customization to different acoustic and application contexts. The concept of acoustic context comprises all the elements that, in different ways, affect the speech signal captured by the microphone/s deployed in the environment. These elements depend not only on the user but also in the physical environment surrounding the speaker. On the other hand, the concept of application context refers to the semantic structure of the domains in which the dialog takes place.

The objectives of the current project imply the need to develop strategies to allow characterizing the working conditions of the dialog system (acoustics, speech style, semantic context, user type, etc.) and defining and implementing adaptation techniques for such conditions. The use of adaptation and customization techniques will lead to significant changes in the traditional evaluation and usability metrics used, which will be addressed in the project. This project will also extend the previously developed DIHANA dialog system to new application contexts, allowing the user to achieve multiple objectives during the dialog transactions.

In the framework of the evaluation of the system's usability and operation, we pretend:

- The search of the appropiate parameters for the evaluation of the developed technologies.
- A comparative evaluation of the designed strategies and the evaluation of the adaptation of the different components in our dialog system.
- An analysis of the results.

Finally, within EDECAN, we will build and evaluate a fully working prototype of a dialog system on different scenarios.

5. Conclusions

A complete dialog system for information access using spontaneous speech in a restricted domain task has been presented in this work. Different modules perform specific functionalities in order to carry out the final goal of the system.

The basic characteristics of these modules have been presented throughout the article. Different methodologies and strategies to carry out its specific functionality have been mentioned. Most of these methodologies are based on the data corpus adquired for the DIHANA project taking into account the users opinion about the system perfomance.

Error detection and correction techniques have also been developed. These techniques allow us to distinguish the situations in which errors appear and to make the necessary corrections to satisfactorily complete the task.

The results obtained in the evaluation of the different components, as well as in the evaluation of the global system, are quite satisfactory. However, new methodologies are being developed to consider the appropiate parameters for the evaluation of the different modules and to determine whether the system behaviour is correct or not when the modules have been adapted to carry out different tasks. These works are currently being developed in a new project called EDECAN.

6. Acknowledgements

Work partially supported by the Spanish CICYT under contract TIN2005-08660-C04-02 and the GVA project ACOMP06/053.

7. References

[1] J.M. Benedí, A. Varona, and E. Lleida, "DIHANA: Sistema de diálogo para el acceso a la información en habla espontánea en diferentes entornos," in *Actas de las III Jornadas en Tecnología del Habla*, Valencia (España), 2004, pp. 141–146.

[2] A. Bonafonte, P. Aibar, E. Castell, E. Lleida, J. B. Mariño, E. Sanchis, and M. I. Torres, "Desarrollo de un sistema de diálogo oral en dominios restringidos," in *I Jornadas en Tecnología del Habla*, Sevilla (Spain), 2000.

[3] "Sphinx Website," http://www.speech.cs.cmu.edu/sphinx/.

[4] Philip Clarkson and Ronald Rosenfeld, "Statistical Language Modeling Using the CMU–Cambridge Toolkit," in *Proc. Eurospeech'97*, Rhodes, (Greece), 1997, pp. 2707–2710.

[5] E. Segarra, E. Sanchis, F. García, and L.F. Hurtado, "Extracting semantic information through automatic learning techniques," in *International Journal of Pattern Recognition and Artificial Intelligence*, Salt Lake City (USA), 2002, pp. 16(3):301–307.

[6] F. Torres, L.F. Hurtado, F. García, E. Sanchis, and E. Segarra, "Error handling in a stochastic dialog system through confidence measures," in *Speech Communication*, 2005, pp. (45):211–229.

[7] D. Griol, F. Torres, L.F. Hurtado, E. Sanchis, and E. Segarra, "Different approaches to the dialogue management in the DIHANA project," *10th International Conference SPEECH and COMPUTER (SPECOM)*, pp. 203–206, 2005.

[8] F. García, L.F. Hurtado, E. Sanchis, and E. Segarra, "The incorporation of Confidence Measures to Language Understanding," in *International Conference on Text Speech and Dialogue (TSD 2003). Lecture Notes in Artificial Intelligence series 2807*, Ceské Budejovice (Czech Republic), 2003, pp. 165–172.

[9] R. San-Segundo, B. Pellom, K. Hacioglu, W. Ward, and J. Pardo, "Confidence measures for spoken dialogue systems," in *Proc. ICASSP*, Salt Lake City (USA), 2001.

[10] T. Hazen, S. Seneff, and J. Polifroni, "Recognition confidence scoring and its use in speech understanding systems," in *1997*, 2002, pp. 16, 49–67.

[11] "EDECAN Project website: Multidomain Dialog System with Acoustic and Application Adaptation," http://www.edecan.es/.