A hybrid approach to the development of dialogue systems directed by semantics

Emilio Sanchis, Isabel Galiano, Fernando García, Antonio Cano
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46020-Valencia, SPAIN
esanchis,mgaliano,fgarcia,acano@dsic.upv.es

Abstract

In this work we present an approach to the development of the BASURDE[1] dialogue system, which answers telephone queries about railway timetables in Spanish. We will focus on the understanding and dialogue components which are modeled under a stochastic framework. The preliminary results from semantic and dialogue interpretations of user dialogue turns are also included in this work.

1 Introduction

In the development of dialogue systems many knowledge sources must be taken into account. The specific characteristics of each knowledge source imply that different kinds of models and different architectures can be used. It is widely accepted that stochastic models are a good representation for some of these knowledge sources and some specific works have been done to represent the semantic of the sentences and the dialogue structure (Pieraccini *et al.*, 1997)(Baggia *et al.*, 1999)(Lamel *et al.*, 2000)(Martinez *et al.*, 2000)(Segarra *et al.*, 2001).

We present an approach in which the dialogue structure is represented by a stochastic network of dialogue acts. One advantage of this kind of network is that it can be learnt from annotated training samples. Moreover, it gives us a prediction of the next dialogue acts which are expected from the user as well as some information about the possible dialogue acts that can be generated by the system. The identification of the user dialogue acts is done through the semantic representation of the sentence. This semantic interpretation not only supplies the corresponding dialogue act but also supplies the information given about the query constraints, such us Date, Departure_city, etc.

To be able to provide the information requested by the user, the system has to manage the values supplied by the user during the conversation. We do this by means of a record of current values that is updated after each user turn and is used to generate the database queries and to participate in the generation of the dialogue turns of the system.

2 The Dialogue module

The dialogue model proposed is a stochastic network which is automatically learnt from a training set of dialogue samples obtained by the Wizard of Oz technique (Figure 1). A dialogue sample is a concatenation of dialogue acts which represents the translation of a given user utterance into a sentence of a dialogue act language.

One important decision is the definition of the set of dialogue acts associated to the application. If we establish a low number of dialogue acts that are independent from the task, we can expect a good modelization of the dialogue structure and an easy identification of the dialogue acts which are generated by the user; we could also change the application without having to make many changes in the dialogue model. However, in order for the system to generate its dialogue turn, more in-

[1]Work partially funded by *CICYT* under project TIC98-0423-C06

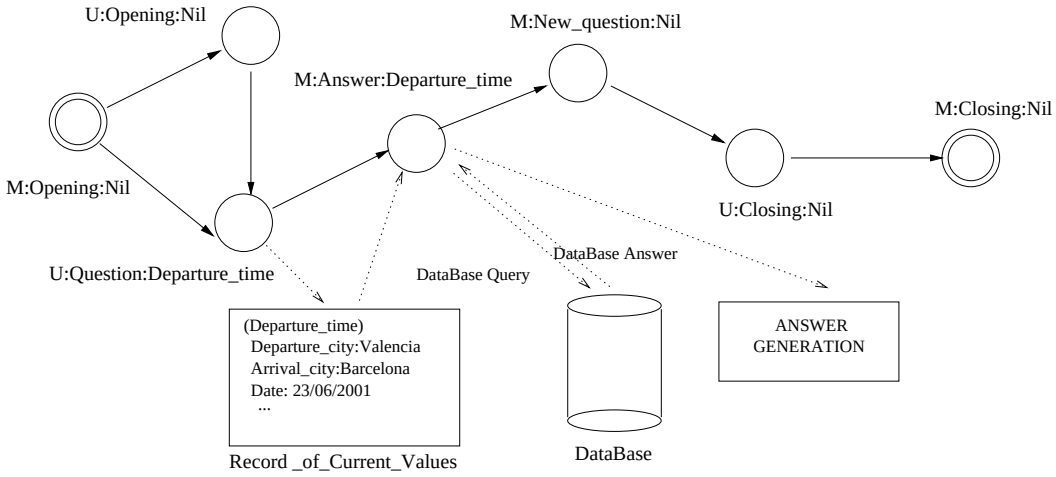

Figure 1: *An example of a part of the Dialogue model.*

formation about the content of the sentences is required.

If we increase the number of dialogue acts so that each dialogue act has a more specific meaning, then the variability of decisions (or actions) associated to each state in the network is reduced. In other words, a dialogue act will have a very specific intention, but we will need a huge number of labeled dialogues to learn the model. For example, if the act is *Question* there are many kinds of questions associated to it, but if the act is *Question:Departure_time* it only represents a question about the departure time.

Since a balance between the number of labels and the model structure is needed, within the BASURDE project we have defined a set of three-level dialogue acts. This set represents not only information about a general dialogue behaviour but also information about the task (Martinez *et al.*, 2000).

The first level of each dialogue act labels the dialogue behaviour. The labels we define at this level are generic for any task. The second level is related to the semantic representation of a sentence and it is specific to the task. In the Dialogue model presented here only the first two levels are considered.

The following labels have been defined for the first level: *Opening, Closing, Undefined, Not_understood, Waiting, Affirmation, Rejection, Question, Confirmation, Answer.* The labels defined for the second level are: *Departure_time, Return_departure_time, Arrival_time, Return_arrival_time, Price, Departure_city, Arrival_city, Lenght_of_trip, Stops, Departure_date, Arrival_date, Train_type, Services.*

For example, a dialogue turn can be labeled as follows:

Me puede decir el horario de los trenes a Valencia el próximo lunes ?
(Can you tell me the timetable to Valencia for next Monday ?)
(Question: Departure_time)

The stochastic network representing the Dialogue model is obtained from a training set of dialogues which are labeled in terms of dialogue act sequences. This network is built by using the bigram probabilities.

The dialogue act network can be used in two ways:
- To predict the next dialogue act of the user; helping the recognition and understanding processes.
- To decide the next action of the system. As there are not enough samples to learn an accurate model this decision making process should be driven by the semantics.

Now we will describe how the dialogue manager works. It has two main components: the dialogue network and the record of current values. The input of this module is supplied by the Understanding module. This input is a frame representation of the semantic information obtained from the user turn. We can extract the corresponding di-

alogue act from each input frame as well as the constraints about the query given by the user. The Dialogue Manager uses this information in two ways: it determines the next dialogue transition to be made and updates the record of current values using the constraints obtained from the query.

The Dialogue Manager output, which is also a frame representation, is sent to the answer generator and then to the synthesizer.

The dynamics of this process is given by the following Dialogue Manager algorithm:

```
/*Initialization*/
Put State=Opening
Init(Record_of_Current_Values) /*Init(RCV)*/
Repeat
   Sentence=obtain sentence from the user turn
   Frame=extract_meaning(Sentence)
   State=Transition_to(State,Frame)
   RCV=Update(Frame)
   /* actions of the manager */
   if  complete_query(RCV)
     then
       Send_Database_query
       State=Choose_transition
     else
       select transitions permitted by RCV
       State=Choose one of these selected transitions
   Generate output frame
until State=Closing
```

The dialogue manager accepts the frames obtained from the user turn as input. First it modifies the record of current values if necessary. If there is enough information in this record, a query to the database is made, an output frame with the answer is generated, and a transition in the dialogue network is made. Otherwise the record of current values is used to determine which transitions of the dialogue network should be pruned, i.e. those that are not compatible with the updated information. This situation occurs because the model is learnt from a limited set of samples and it is a bigram model with just one label history, and then the constraints given in previous turns can not be taken into account.

For example, one of the transitions of the network might imply asking the user about the departure city and this information has already been given in a previous turn. In this case the corresponding transition would be forbidden. After the set of allowed transitions is determined, one of them is selected and the corresponding output frame is generated. The process finish when a *Closing* label is found.

3 The Understanding module

We use frames to represent the meaning of sentences. Each frame represents a concept and can have some attributes associated to it. We have defined 18 types of frames; some of them are related to the task (for example *Departure_time*, *Price*, etc.), and others are related to the general characteristics of the dialogues (for example, *Not_understood*, *Affirmation*, etc.). Note that each type of frame has a corresponding dialogue act associated to it.

We use a two phases approach for the understanding process (Figure 2). In the first phase a sequence of semantic units and its corresponding segmentation is obtained from an input sentence. In the second phase one or more frames are extracted from this semantic sentence.

The first phase is implemented from a stochastic transduction point of view (Segarra *et al.*, 2001); that is, the input sentence (in words) is translated into an output sentence of a semantic language. The vocabulary of this semantic language is composed by a set of semantic units, that represents meanings of segments of words.

The segmentation of these sentences allows us to define two kind of stochastic models: the Semantic model and the Semantic-Unit model. The Semantic model represents the allowed sequences of semantic units and their probabilities. The Semantic-Unit model represents the allowed sequences of words and their probabilities which are associated to each semantic unit.

These models can be automatically learnt from a set of annotated training samples. In this work, we have used bigram models, but any other type of stochastic model, like n-grams or automata learnt by Grammatical Inference techniques (Segarra *et al.*, 2001), could be used. These models can be integrated into a unique understanding model; each semantic unit of the Semantic model

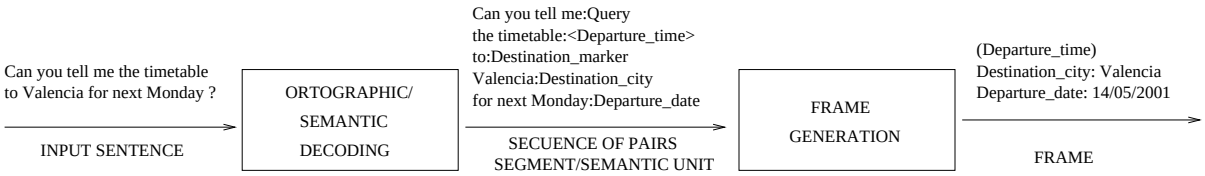

Figure 2: *Transduction approach in two phases.*

is substituted by its corresponding Semantic-Unit model.

As we already have the Dialogue model of the system, we can obtain a specific Semantic model for each user dialogue act. However, the use of specific models can lead to a lack of training samples and therefore to the use of tied models. In our case, we have defined only specific models for the first level of our set of dialogue labels.

A Viterbi decoding algorithm supplies the most likely sequence of semantic units for the input sentence and its corresponding segmentation. As the output of the Understanding module is a frame, a set of rules is applied to obtain the corresponding frame. These rules permit us to leave behind semantic units such as courtesy, markers, etc., and select only the semantic units that have a corresponding slot associated to the frame.

4 Experimental Results

In this section we report the preliminary results from semantic and dialogue interpretations of user dialogue turns.

We defined a training set of 175 dialogues with 1,141 user utterances and a test set of 40 dialogues with 268 user utterances from the orthographic transcription of a set of 215 dialogues, obtained through a Wizard of Oz technique. The number of words in these two sets was 11,987 and the medium length of the utterances was 10.5 words. The percentage of correctly understood sentences (correct frames) was 80%, and the percentage of correct user dialogue act identification was 87%.

5 Conclusions

We have presented an approach for the development of dialogue systems based on stochastic models which are automatically learnt from training samples. We have proposed a system architecture which includes an Understanding module that extracts the semantics of the user turns in terms of frames. A preliminary implementation of the system has been done, and preliminary results are reported. We hope that the behaviour of the system improves when we have more dialogue training samples. We will model other dialogue situations which were not encountered in our current training corpus.

References

Baggia P., Kelner A., Pérennou E., Popovici C., Strum J., Wessel F. 1999. Language Modeling and Spoken Dialogue Systems the ARISE experience. *Eurospeech99*, pp. 1767–1770.

Bonafonte A., Castell N., LLeida E., Mariño J.B., Sanchis E., Torres M.I., Aibar P. 2000. Desarrollo de un sistema de diálogo oral en dominios restringidos. *I Jornadas en Tecnología del Habla, Sevilla.*

Lamel L., Rosset S., Gauvain J.L., Bennacef S., Garnier-Rizet M., Prouts B. 2000. The LIMSI Arise system. *Speech Communication*,31, pp. 339–353.

Martinez C., and Casacuberta F. 2000. A pattern recognition approach to dialog labelling using finite- state transducers. *V Iberoamerican Symposium on Pattern Recognition*, pp. 669–677.

Pieraccini R, Levin E., and Eckert W. 1997. AMICA: the AT&T Mixed Initiative Conversational Architecture. *Eurospeech97*, pp. 1875–1878.

Segarra E., Sanchis E., Galiano M., García F., Hurtado L.F. 2001. Extracting semantic information through automatic learning techniques. *IX Spanish Symposium on Pattern Recognition and Image Analysis (AERFAI), Castellón.*