A KNOWLEDGE REPRESENTATION MODEL FOR A VOICED DIALOGUE SYSTEM

R. López-Cózar, A. J. Rubio, P. García, J. Díaz
Dpto. Electrónica y Tecnología de Computadores
Universidad de Granada, 18071 Granada, España (Spain)
Tel.: +34-58-243193, FAX: +34-58-243230, e-mail: rubio@hal.ugr.es

ABSTRACT

We present a preliminary version of a voice dialogue system suitable to deal with user orders and questions in fast-food restaurants. The system, called SAPLEN (Sistema Automático de Pedidos en LEnguaje Natural, *Automatic System for Natural Language Ordering*), consists of two main sub-systems, namely a dialogue sub-system and a voice interface **[1]**, **[2]**, **[3]**. The dialogue sub-system is a natural language processing system that may be considered a rule-based expert system, whose behavior is decided from a recorded dialogue corpus obtained at a real restaurant. In this paper we will describe the voice interface and the dialogue sub-system, and will focus on knowledge representation for the dialogue sub-system. Finally, we will show a dialogue example carried out between the system and an imaginary user.

1. THE VOICE INTERFACE

The voice interface is used to send information from the user to the dialogue sub-system and vice-versa. We are using a continuous-speech recognition system **[4]** which is carried out on a workstation. The main characteristics of the speech recognizer are the following:

- Sampling frequency: 8 KHz, 8bit, mu-law.
- The recognition system includes a voice activity detector, which is trained in a discriminative manner to distinguish between voice and background noise.
- The speech signal is pre-emphasized and segmented into frames 30 ms. long. The frames are overlapped and the resultant frame period is 10 ms. Every frame is analyzed and represented by a vector including 14 Mel Frequency Cepstral Coefficients, the energy and the first and second derivatives.
- We are using context-independent phone-like units, which are modeled by SCHH (Semi-Continuous Hidden Markov Models).
- The language is modeled by a bigram. The vocabulary size is 250 words. The perplexity of the bigram is not evaluated at this stage of the setting up, as now we are not considering real conversations to estimate the probabilities, but only to establish the vocabulary and to set up the dialogue system.

2. THE DIALOGUE SUB-SYSTEM

The system goal is to simulate the restaurant-clerk behavior. It must be able to provide information and ask user questions similarly to how a human clerk does. In addition we want it to process spontaneous voiced-speech, which at a linguistic level means to take into account phenomena such as unnecessary word repetition, grammatical order change, anaphora, discordances, context information, grammatical mistakes, etc. Some more information about these kinds of systems can be found in **[5]**, **[6]**, **[7]**.

3. KNOWLEDGE REPRESENTATION MODEL

The dialogue sub-system uses several kinds of knowledge, which are represented as frames, rules and class instances.

3.1 Frames

We use frames to represent the user interactions **[8]**. A frame is a compound of a slot set. Each slot has associated values and possible value restrictions. We use other frame type to represent context information, which can be added to the users' natural language utterance. In our system, an user interaction can generate one or more frames, and we have defined restrictions for the possible values of the slots. Each frame consists of two parts, one of them is used to store the speech act type, and the another one is used to store restaurant product information. In our model we use four speech act types to represent the user intentions: *"greeting"*, *"order"*, *"question"* and *"modification"*, that are included in the user interaction representation. The frames used to store interactions related to foods have slots to store the amount of food wanted, food type, ingredients and optional complements. The frames used for drinks have slots to store the amount of drink, taste, size and optional complements.

3.2 Rules

We represent the knowledge necessary for the users' natural language analysis as syntactic and semantic rules. Three syntactic rules used are shown in the next table.

<restaurant_product> ⇒ <food> | <drink>
<food> ⇒ <food_type> <food_name> [<food_complements>
<drink> ⇒ <drink_name> <taste> <size> [<drink_complements>]

Table 1. Three syntactic rules.

These rules tell the system that a restaurant product can be either a food or a drink. Any food belongs to type of foods and has a name, some ingredients and possibly some food complements (*ketch-up*, for instance). Any drink has a name, taste, size, and possibly some complements (*ice*, for instance).

We show two semantic rules in the next table.

```
<drink_price_question> ::=
{
<interrogative> <price>  <drink_name>
              ||
<drink_name> <interrogative> <price>
              ||
<drink_name> <price> <interrogative>
}


<available_food_question> ::=
{
<interrogative> <food> <existence>
              ||
<interrogative> <existence> <food>
              ||
<food> <interrogative> <existence>
}
```

Table 2. Two semantic rules.

The <drink_price_question> semantic rule tells the system what it must *understand* as an user's desire to know the price of some drink. The <available_food_question> tells the system what it must *understand* as an user's desire to know the foods that are available in the restaurant.

We use a semantic grammar for the analysis of the user utterance **[9]**, **[10]**, **[11]**. A semantic grammar is a free-context grammar in which the choice of both non terminals and rules is deciced from both syntactic and semantic considerations. Semantic rules are usually associated to syntactic rules. The first are usually written taking into account semantic key concepts. This kind of analysis combines morphological, syntactic and semantic analysis -as well as discourse integration- in one process. Its main advantages are two, on the one hand, many syntactic ambiguities can be avoided in case they have no semantic meaning, whilst on the other, syntactic details that have no effect in semantic analysis can be ignored.

Another type of linguistic knowledge is used to generate natural language words and sentences. The next tables show some rules used to generate Spanish words and sentences.

<word> ⇒ <word_stem> <gender> <number>
<gender> ⇒ **<O>** | **<A>**
<number> ⇒ < > | **<S>** | **<ES>**

Table 3. A rule for Spanish word generation.

<inform_introduction> <article> <product> <cost> ^{C} <product_price> <money> ^{C}

where:

<inform_introduction> ⇒ **<PUES><,>** | **<VAMOS A VER><,>** | **< >**

<article> ⇒ **<EL>** | **<LA>** | **<LOS>** | **<LAS>**

<product> ⇒ <food> | <drink>

<food> ⇒ [article] [<amount>] <food_type>^{C} <food_name> [<food_complements>]

<drink> ⇒ <article> [<amount>] <drink_name>^{C} <size> [<taste>] [<drink_complements>]

<cost> ⇒ [**<ALSO>**] { **<COSTS>** | **<COST>** }

<product_price> ⇒ look_for_price(<drink_name>,<size>)

<money> ⇒ **<PESETAS>**

<x> ^{C} ⇒ <x> | <>

Table 4. A rule for Spanish sentence generation.

In *Table 4* <x> ^{C} represents contextual information. It means that the information in <x> can be either included or omitted from the natural language generation depending on contextual considerations. The system uses this rule to inform the user about the price of products. For instance, a sentence it can generate is: "*el bocadillo de jamón con queso cuesta 350 pesetas" (the jam and cheese sandwich costs 350 pesetas).*

The system uses some strategic knowledge stored as rules in order to sell restaurant products. The next table shows one of these rules.

```
<target_product_selling> ::=
        if {!tried_to_sell(food) and
           !tried_to_sell(drink)
           }
               try_to_sell(food) and try_to_sell(drink)
        otherwise
        if {tried_to_sell(food) and
           !tried_to_sell(drink)}
           }
               try_to_sell(drink)
        otherwise
        if {!tried_to_sell(food) and
            tried_to_sell(drink)
           }
               try_to_sell(food)
        otherwise
        if {tried_to_sell(food) and
           tried_to_sell(drink) and
           !tried_to_sell(more)
           }
               try_to_sell(more)
```

Table 5. An strategic rule.

The <target_product_selling> rule tells the system that it must try to sell a food and a drink, at least. When this is done, it must try to sell more products.

3.3 Class instances

The information regarding restaurant products is located in one module of the system called Restaurant-product Knowledge Base. This information is stored as instances of the classes FOOD, INGREDIENT, FOOD_COMPLEMENT, DRINK, SIZE, TASTE, DRINK_COMPLEMENT and OTHER_PRODUCT defined as it is shown in the next tables.

<food_type>
<food_name>
<price>
<temperature>
<ingredients>
<available_complements>
<availabe_complements_prices>

Table 6. Food class.

<drink_name>
<size>
<available_tastes>
<price>
<temperature>
<alcoholic-carbonated>
<available_complements>
<available_complements_prices>

Table 7. Drink class.

<ingredient>
<ingredient_gender>
<ingredient_conector>
<inferred_food>

Table 8. Ingredient class.

<food_complement>
<food_complement_gender>

Table 9. Food-complement class.

<size>
<size_conector>

Table 10. Size class.

<taste>
<taste_conector>
<inferred_drink>

Table 11. Taste class.

<drink_complement>
<drink_complement_gender>

Table 12. Drink-complement class.

<other_product_gender>
<other_product_type>

Table 13. Other-product class.

It is assumed that restaurant managers are allowed to change the information in the Restaurant-product Knowledge Base, to add, update or remove available products. This fact implies the possibility of setting inconsistent information, and then cause the system's malfunction. One module of the system, called Consistency-Check Module, aims to avoid this risk by auto-checking the information consistency when booting the system (for instance, the gender and price of every product must be defined). Inconsistency is forbidden in the system, so in case any missed or inconsistent information is detected, the system reports it so and the system function is halted.

4. DIALOGUE EXAMPLE

Next we show an example of dialogue (translated from Spanish to English) between the system and an imaginary user. Note that the answers of the user to the system's questions are unexpected sometimes. This is done to show the fact that though the dialogue management is system-directed, the user can take the initiative in the dialogue whenever he/she wants. If the user answers a question unexpectedly it causes a focus shifting in the conversation that needs to be handled by the system. To do so, the system can reference restaurant products differently, depending on the actual focus of the conversation. The words within brackets represent system confirmations of user orders.

SYSTEM: *GOOD EVENING, WELCOME TO OUR AUTOMATIC PRODUCT-ORDERING SYSTEM., WHAT WOULD YOU LIKE TO HAVE?*
USER: *WHAT IS THE PRICE FOR A TUNA SANDWICH AND TWO MEDIUM BEERS*
SYSTEM: *WELL, LET'S SEE, TUNA SANDWICH COSTS 150 PESETAS, AND TWO MEDIUM BEERS COST 500, WOULD YOU LIKE TO HAVE SOMETHING?*
USER: *A HAM SANDWICH AND SEVERAL BEERS*
SYSTEM: *(SEVERAL BEERS) SORRY, HAM SANDWICHES ARE NOT AVAILABLE, WE HAVE TUNA, OMELETTE AND SQUID, WOULD YOU LIKE ANYTHING ELSE?*
USER: *I ALSO WANT THREE MILKSHAKES*
SYSTEM: *(3 MILKSHAKES) WOULD YOU LIKE CHOCOLATE OR VANILLA?*
USER: *WHICH SALADS DO YOU HAVE*
SYSTEM: *WE HAVE TROPICAL AND MIXED, WHICH ONE WOULD YOU LIKE?*
USER: *A MIXED ONE*
SYSTEM: *(1 SALAD MIXED) WHICH FLAVOUR WOULD YOU LIKE FOR THE MILKSHAKES? CHOCOLATE OR VANILLA?*
USER: *WELL A CHOCOLATE ONE AND TWO VANILLA*
SYSTEM: (1 MILKSHAKE CHOCOLATE) (2 MILKSHAKES VANILLA) *WHICH SIZE FOR VANILLA MILKSHAKE? LARGE OR SMALL?*
USER: *MEDIUM*
SYSTEM: *SORRY, WE DO NOT HAVE MEDIUM VANILLA MILKSHAKES, WHICH OTHER SIZE WOULD YOU LIKE? LARGE OR SMALL?*
USER: *SMALL*
SYSTEM: *(2 MILKSHAKES VANILLA SMALL) WHICH SIZE WOULD YOU LIKE FOR CHOCOLATE MILKSHAKE? SMALL OR MEDIUM?*
USER: *WHICH SANDWICHES DO YOU HAVE*
SYSTEM: *WE HAVE TUNA, OMELETTE AND SQUID, DO YOU WANT ONE?*
USER: *WHAT IS THE PRICE FOR TUNA ONES*
SYSTEM: *TUNA COSTS 150 PESETAS, DO YOU WANT ONE?*
USER: *YES ONE*
SYSTEM: *(1 SANDWICH TUNA) WHICH SIZE WOULD YOU LIKE FOR CHOCOLATE MILKSHAKE? SMALL OR MEDIUM?*
USER: *SMALL ALSO*
SYSTEM: *(1 MILKSHAKE CHOCOLATE SMALL) WHICH IS THE SIZE FOR THE BEERS? SMALL OR MEDIUM?*
USER: *MEDIUM*
SYSTEM: *(BEERS MEDIUM) HOW MANY?*
USER: *THREE*
SYSTEM: *(3 BEERS MEDIUM) ANYTHING ELSE?*
USER: *NO*
SYSTEM: *LET'S SEE, YOU ORDERED FOR 1 MIXED SALAD, 1 TUNA SANDWICH, 1 CHOCOLATE SMALL MILKSHAKE, 2 VANILLA SMALL MILKSHAKES, AND 3 SMALL BEERS. 1500 PESETAS PLEASE. THANKS FOR USING OUR AUTOMATIC PRODUCT-ORDERING SYSTEM. GOOD EVENING.*

5. REFERENCES

[1] Ramón López-Cózar, Pedro García, J. Díaz, Antonio J. Rubio. "A Voice Activated Dialog System for Fast-Food Restaurant Applications". Proc. Eurospeech'97.

[2] Ramón López-Cózar Delgado, Antonio J. Rubio Ayuso. "Una Introducción al Mecanismo de Generación de Lenguaje Natural utilizado por el Sistema SAPLEN". Proc. XIII SEPLN Conference.

[3] Ramón López-Cózar Delgado, Antonio J. Rubio Ayuso. "SAPLEN: Un Sistema de Diálogo en Lenguaje Natural para una Aplicación Comercial". Proc. III Jornadas de Informática.

[4] L. Rabiner, B.H. Juang. "Fundamentals of Speech Recognition". Prentice-Hall, 1993.

[5] K. Hatazaki, F. Ehsani, J. Noguchi, T. Watanabe. "Speech dialogue system based on simultaneous understanding". Speech Communication 15 (1994) 377-378

[6] V. Zue, S. Seneff, J. Polifroni, M. Phillips, C. Pao, D. Goodine, D. Goddeau, J. Glass. "PEGASUS: A spoken dialogue Interface for on-line air travel planning". Speech Communication 15 (1994) 377-378

[7] S. Seto, H. Kanazawa, H. Shinchi, Y. Takebayashi "Spontaneous Speech dialogue system TOSBURG II and its evaluation". Speech Communication 15 (1994) 377-378

[8] "Inteligencia Artificial, Segunda edición". Mc-Graw Hill, 1994.

[9] Burton, R. R. "Semantic Grammar: An engineering technique for constructing natural language understanding systems". Tech. Rep. 3453, Bolt Beranek and Newman, Boston.

[10] Hendrix G., Sacerdoti E. D., Sagalowicz D., Slocum J. "Developing a natural language interface to complex data". ACM transactions on data base systems 3, 105-147.

[11] Hendrix G., Lewis W.H. "Transportable natural-language interfaces to databases". 19th annual meeting of the association for computational linguistic.