A VOICE ACTIVATED DIALOGUE SYSTEM FOR FAST-FOOD RESTAURANT APPLICATIONS

R. López-Cózar, P. García, J. Díaz and A.J. Rubio
Dept. 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 client orders and questions in fast-food restaurants. The system consists of two main sub-systems, namely a dialogue sub-system and a voice interface. The dialogue sub-system is a natural language processing system that may be considered a rule-based expert system, whose behaviour is decided from a recorded dialogue corpus obtained at a real restaurant. In this paper we present a general description of both sub-systems, and focus on knowledge representation, grammar, and module structure of the dialogue sub-system. An introduction to the natural language generation mechanism used is introduced, and future work is mentioned. Finally some conclusions are shown.

1. INTRODUCTION

Applications of Speech Technology include all kinds of systems in which a part of the communication process is carried out by the voice. Real-world applications involve a human being trying to communicate to a machine to get some information or service. Relatively simple Automatic Speech Recognition systems can be used to achieve some goals when the dialogue is heavily limited, as it is usually the case of isolated-word speech recognition systems. The dialogue restrictions imply training and collaboration on the part of the users. Unrestricted dialogue applications are much more appealing as users do not need to be trained and collaboration requirements are minimal. These systems include a dialogue module that is an important part of the whole system.

In this paper we are presenting a first version of a voice activated dialogue system intended to be used in fast-food restaurants, probably to deal with phone ordering. The system we present is basically a compound of a voice interface and a dialogue sub-system. Next we explain the main characteristics of both sub-systems.

2. THE VOICE INTERFACE

The voice interface is used to send information from the client to the dialogue sub-system and vice-versa. At the present stage of the system, the output of the dialogue sub-system is presented directly on the screen. We plan to use a Spanish Text-to-Speech converter in the near future. The speech input to the system has to be converted to text in order to be processed by the dialogue module. We are using a continuous-speech recognition system **[1]** 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 modelled by SCHH (Semi-Continuous Hidden Markov Models).
- Language is modelled by a bigram. Vocabulary size is 250 words. Perplexity of the bigram is not evaluated at this stage of the setting up, as now, we are not considering real conversations to estimate probabilities, but only to establish vocabulary and to set up the dialogue system.

3. THE DIALOGUE SUB-SYSTEM

The system goal is used to simulate restaurant-clerk behaviour. It must be able to provide information and ask client 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. We also expect a learning ability for the system to allow new information (foods, drinks, ingredients, etc.) acquisition from client interaction. Though the system always attempts to initiate the conversation with the clients (system-directed dialogue) they are free to answer a question with another question, i.e., they are allowed to act unexpectedly (focus shifting) **[2]**. Some more information about these kinds of systems can be found in **[3]**, **[4]**, **[5]**, **[6]**.

4. KNOWLEDGE REPRESENTATION

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

4.1 Frames

We use frames to represent client interaction. Each frame represents a class of elements, and is a compound of a slot set. Each slot has associated values and possible value restrictions. When necessary slots in one frame are filled, it represents a class instance (**[7]**, **[8]**). To represent client intentions we use four speech act types: *"greet"*, *"order"*, *"question"* and *"modification"* as pragmatic information to include in client interaction representation (**[9]**,**[10]**). Another frame type is used to represent context information, which can be added to the clients' natural language utterance.

4.2 Rules

We represent the linguistic knowledge necessary for clients' natural language analysis as syntactic and semantic rules, which are stored in the Output Interface Knowledge Base. Another type of linguistic knowledge, stored as rules in the Output Interface Knowledge Base, is used to generate natural language words and sentences. The system also uses some strategic knowledge in order to sell restaurant products. This knowledge is stored as rules in the Initiative Module (sub-module of the Control Module).

4.3 Class instances

The information regarding available products in the restaurant is stored in the Restaurant-product Knowledge Base, as instances of classes FOODS and DRINKS defined properly. We use other class instances to represent unavailable products in the restaurant.

5. GRAMMAR

We use a semantic grammar for dialogue sub-system analysis (**[7]**,**[11]**,**[12]**,**[13]**). A semantic grammar is a free-context grammar in which the choice of both non terminals and rules is decided from syntactic and semantic considerations. Semantic rules are usually associated to syntactic rules. The first are usually written considering semantic key concepts. The main advantages of this kind of analysis 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.

6. MODULE STRUCTURE

The dialogue sub-system is a compound of modules described below:

- Input Interface
- Control Module
- Memory Module
- Restaurant-product Knowledge Base
- Lexicon
- Output Interface

6.1 Input interface

The natural language client utterance enters the Input Interface where it is converted into its semantic interpretation, which is sent to the Control Module **[14]**. Modules in the Input Interface are:

- Context Module
- Natural Language Processor
    - Syntactic Analyzer
    - Semantic Analyzer
- Input Interface Knowledge Base
    - Syntactic-rule Knowledge Base
    - Semantic-rule Knowledge Base

Semantic interpretation is obtained from keywords by both syntactic and semantic rules. In case semantic interpretation cannot be built up because of unresolved ambiguity, unrecognized words or unresolved references in the client utterance, the system informs the client about this fact and asks him/her to either clarify the utterance or enter it again differently.

6.2 Control module

The Control Module uses the semantic interpretation from the Input Interface and feeds the Output Interface with activate signals for the Natural Language Generator. It also feeds the Memory Module with the understood new information. We expect it could update the Restaurant-product Knowledge Base as new restaurant products are learned (not yet set up).

Modules in the Control Module are as follows:

- Product-orders Control Module
- System-actions control Module
- Learning Module
- Initiative Module
- Non-understanding Module
- Client-greeting Module
- Client-order Module
- Client-question Module
- Client-modification Module
- System-identification Module
- Consistency check Module

6.3 Memory Module

The Memory Module receives the understood new information from the Control Module and sends back dialogue history information.

Modules in the Memory Module are:

- Memory
- Frame Maker
- Frame Organizer
- Frame Unifier
- Slot Modifier

The Memory stores all dialogue history information. The Frame Maker gets the understood new information from the Control Module and makes the necessary frame amount in the Memory to store it. The Frame Organizer sorts out frames by putting invalid ones together into a group, which is moved to a specific position in the Memory. The Frame Unifier stores the understood new information in the uncompleted product orders, by means of frame-unification. We expect the Slot Modifier will receive the understood new information from the Control Module and modify the effected slots because of changes in client desire (not yet carried out).

6.4 Restaurant-product Knowledge Base

The Restaurant-product Knowledge Base stores restaurant available product (foods and drinks) information, and information about unavailable products that are usually ordered in other restaurants. As mentioned above, we expect to carry out a learning mechanism for client interaction so that new information can be added.

6.5 Lexicon

The Lexicon stores information about other keywords necessary for syntactic and semantic analysis. These keywords (interrogatives, negatives, affirmatives, etc.) are also grouped into classes.

6.6 Output Interface

The Output Interface sends to screen the new system response in natural language. Modules in the Output Interface are:

- Reference-control Module
- Natural Language Generator
    - Inform Generator
    - Question Generator
    - Greeting Generator
- Ticket Generator
- Output Interface Knowledge Base
    - Word-form Rule Knowledge Base
    - Sentence-form Rule Knowledge Base
    - Greeting Knowledge Base
    - System-identification Knowledge Base

The Reference-control Module sends information to the Inform Generator and to the Question Generator about the time-distance to the reference (food or drink). The Natural Language Generator provides the next natural language system response from the activate signals provided by the Control Module. These signals decide which Natural Language Generator modules (one or more) must be activated. The Ticket Generator continuously provides the list of ordered products on the top right corner of the screen.

7. THE NATURAL LANGUAGE GENERATION

The natural language generation (NLG) is carried out in two steps **[15]**. In the first one the system decides what to say (*deep generation*), as in the second one it decides how to say what it has to say (*surface generation*). The surface generation uses the deep generation as input to obtain surface-words and syntactically correct sentences.

7.1 The deep generation

The system uses three speech act types to obtain the NLG: *"greeting"*, *"inform"*, and *"question"* **[9].** For each, some associated actions are shown below:

<greeting>

<meeting_greeting> <farewell_greeting>

<inform>

<not_understanding> <ambiguity>
<available_products> <product_not_available>
<available_foods> <food_not_available>
<available_contents> <content_not_available>
<available_food_ingredients>
<food_ingredient_not_available>
<available_drinks> <drink_not_available>
<available_sizes> <size_not_available>
<available_flavours> <flavour_not_available>
<available_drink_complements>
<drink_complement_not_available>
<available_product_price>
<ordered_products_price>

<question>

<quantity_slot_fill> <content_slot_fill>
<food_ingredient_slot_fill> <size_slot_fill>
<flavour_slot_fill> <drink_complement_slot_fill>
<food_sell> <drink_sell> <food_and_drink_sell>
<more_sales> <not_understanding_resolve>
<ambiguity_resolve>

7.2 The surface generation

The NLG associated to **<inform>** is carried out by the Inform Generator, that one associated to **<question>** is generated by the Question Generator, and that for **<greeting>** is carried out by the Greeting Generator.

The surface NLG is carried out following two main criteria. On the one hand, the system must include the maximum information in "informs" and "questions", in order to increase the level of client understanding and to decrease the possibility of client confusion, whilst on the other, in order to enhance the level of naturalness, the included information must be minimized by using pronouns and context information available at the moment of the generation. The context is used to avoid repetition or unnecessary inclusion of words.

In order to enhance the level of naturalness, we have carried out a mechanism to simulate the restaurant-clerk behaviour while thinking (*"well, ..."*, *"let's see, ..."*) what to say. For increasing the expression power, we have set up a mechanism of multiple semantic-equivalent questions' assignment to the same system goal. When the system needs to satisfy one of its goals, one of the semantic-equivalent questions is selected in order to satisfy the goal, so that the selected question must always be different to the previous one. The same mechanism is applied for key-concepts: when the system needs a word to express a concept, it selects from among several semantically-equivalent words, so that the selected one must be different to the previous one used to express the same key-concept.

Both the Inform Generator and the Question Generator need to query the Word-form Rule Knowledge Base and the Sentence-form Rule Knowledge Base. The first knowledge base contains rules necessary to obtain surface-words from stem-words. The second one contains rules needed to decide the general form of the sentences, as well as the necessary information to include.

8. FUTURE WORK

Now the dialogue sub-system takes about 30.000 C++ code lines. Its Input Interface is well developed but still needs to define some syntactic and semantic rules. The Output Interface provides a high naturalness level. Only product orders and questions are carried out at the moment, but we are about to start the Modification Module, the Learning Module and the System-identification Module set up. In general, test users have a good opinion about the dialogue sub-system and can get a clear idea about what it should be at the end of its development.

9. CONCLUSION

In this paper we have described the basis we use for the development of the system, its main features and the goals to reach. From a fast-food restaurant dialogue corpus we have decided the set of necessary keywords to analyze clients' sentences, the behaviour for the system and the strategy it must follow. We have used frames, rules and class instances for knowledge representation since they are quite well-suited data structures for these kinds of applications. For sentence analysis we have used a semantic grammar, which combines morphological, syntactic and semantic analysis -as well as discourse integration- in one process. We have shown the module structure of the system as well as the communication between its components. Finally, an introduction to the natural language generation mechanism used has been described, including some mechanisms for enhancing naturalness and expression power.

10. REFERENCES

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

[2] Ferrari, Giacomo. "Towards a realistic dialogue model". SEPLN nº 11, december 1991.

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

[4] 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

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

[6] M. Yamada, F. Itoh, K. Sakai, Y. Komori, Y, Ohora, M. Fujita. "A spoken dialogue system with active/non-active word control for CD-ROM information retrieval". Speech Communication 15 (1994) 377-378

[7] "Inteligencia Artificial, Segunda edicion". Mc-Graw Hill, 1994.

[8] J. Allen. "Natural language understanding". The Benjamin/Cummings Publishing Company Inc., 1995

[9] Susana Nuccetelli. "Reconocimiento de fuerzas ilocutivas en la interaccion Hombre-Maquina". Boletín SEPLN nº 8, diciembre 1989.

[10] M. Nagata, T. Morimoto. "First steps towards statistical modelling of dialogue to predict the speech act type of the next utterance". Speech Communication 15 (1994) 377-378.

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

[12] 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.

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

[14] Irene Castellón, Alicia Manzanera, Antonia Martí. "La interpretación semántica en un sistema de diálogo". Boletín SEPLN nº 8, diciembre 1989.

[15] Iglesias, Carlos A. "Introducción a la generación de lenguaje natural" Revista Informática y Automática, vol. 26-2/1993.