Evaluation of a Multi-user System of Voice Interaction Using Grammars

Elizabete Munzlinger, Fabricio da Silva Soares,
and Carlos Henrique Quartucci Forster

Instituto Tecnológico de Aeronáutica, Divisão de Ciência da Computação,
Praça Marechal Eduardo Gomes, 50 – 12.228-900 São José dos Campos, Brasil
{bety, p2p, forster}@ita.br

Abstract. This paper shows an experimental study about the design of grammars for a voice interface system. The influence of the grammar design on the behavior of the voice recognition system regarding accuracy and computational cost is assessed through tests. With the redesign of a grammar we show that those characteristics can be expressively improved.

Keywords: Grammar, multi-user interface, automatic speech recognition.

1 Introduction

Many speech recognition systems need every new user to train the system to recognize one's voice through the exhaustive reading of texts. This training is necessary because these systems often use extended vocabularies of words [1]. It is desirable to have a system independent of the training and able to recognize the same words when spoken by different voices, with different accents [5]. Applications that use recognized commands don't need such extended vocabulary, which can be restricted to the needs of the particular application. By the use of grammars associated to the application a limit of possible words to every context is determined. The right design of a grammar can make the application become a multi-user system.

The present document shows an experimental study about the design of grammars for a voice interface system for home application (Domotic). The design of grammars based on tests for accuracy and performance analysis made with an ASR (Automatic Speech Recognition) component used to recognize Brazilian Portuguese is described. The knowledge of improved design of grammars is a first step to the automatic generation of a grammar for multi-user interactive applications.

The grammar was used in a prototype of Domotic system that controls up to 32 devices through voice recognition. The system uses the parallel port of the computer and is connected to an electronic circuit that activates the devices. For the ASR system, IBM Via Voice was chosen because its acceptance of Brazilian Portuguese. The Domotic application was developed in Java and uses IBM Java Speech Technology API, which gives access and works together with the IBM VIA VOICE through the JSAPI API [4].

C. Baranauskas et al. (Eds.): INTERACT 2007, LNCS 4663, Part II, pp. 452 – 455, 2007.

2 Grammar Design

A grammar is built from a set of sentences separate by production rules and structured as a tree composed by nodes. The nodes of the grammar are contained in a static structure describing a hierarchy of nodes from the main node and a set of nodes dependent on it. Every node of the grammar has a name who specifies its category [3]. In two-dimensional disposition (Figure 1) it is possible to see the possibilities of connections between the levels of the tree following its hierarchy until reaching the terminal symbols.

At first, we designed a grammar for general use (by systems with several contexts) based on the morphological analysis used in the sentences of the Portuguese language and made of many rules that determines, for example, verbs, subjects, treatments, pronouns and articles. Thus the rule that defines an article comprises other two sub-rules for definite articles and indefinite articles. In the end the grammar has a total of 64 sub-rules and 167 terminal symbols.

It was noticed that this complex grammar lowers the performance of the recognition system making it impossible to execute the application. It took at least 980 MB of memory and 100% of CPU occupancy during 1 minute for allocation and processing of the structure of the grammar. Therefore, there were no computational resources remaining to analyze any sentence.

To solve this problem, the grammar was restructured with changes to the rules composition resulting in the tree showed in Figure 1. The main node of the grammar is the rule *comando* that is composed by the sub-rules, *complemento*, *ação* and *objeto*.

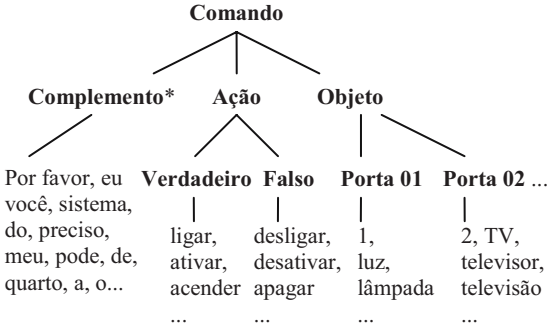

Fig. 1. Composition of rules of the grammar represented by the tree

The rule *ação* has two sub-rules, *verdadeiro* and *falso*, that controls the activation condition of the devices in the Domotic system. The rule *objeto* has one sub-rule for each one of the 32 devices to be controlled, all in the same level of the tree. Like the rule *ação*, this rule also must return the value of accepting of just one of its rules. The sub-rule *complemento* has no value of acceptation and contains 165 terminal symbols extracted from the 35 sub-rules morphologically separated beforehand. Using this grammar, the consumption of memory went down to an average of 423 MB and the duration of total use of the CPU was less than one second. In the new structure of the

grammar the sub-rule *comando* employs the recursivity of the Kleene star operator, permitting 0 to n occurrences of its words in sequence and accepting command variations. Regular grammars have basically the same potentialities of the state machines [2]. Grammar rules can be represented by states of a machine as shown in Figure 2, where R1, R2 and R3 represent the rules *ação*, *objeto* and *complemento* respectively. The recursivity of R3 makes possible the acceptation of any sequences of terminal symbols.

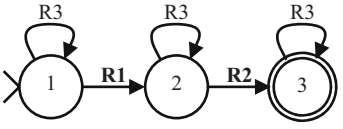

Fig. 2. Grammar represented through a state machine with a recursivity rule

With the recursivity in R3, replicated and interleaved with the other rules, the recognition of simple and complex commands described by the same grammar represented in the Table 1 becomes possible. In Brazilian Portuguese, many complements may appear either in the beginning, middle or in the end of the command (eg.: "*por favor*").

Table 1. Examples of simple and complex commands based in the rules of grammar

Ligar	Luz							
R1	**R2**							
Por favor	eu	preciso	acender	a	lâmpada	do	meu	quarto
R3	R3	R3	**R1**	R3	**R2**	R3	R3	R3
Sistema	você	pode	desligar	o	ventilador	para	mim	por favor
R3	R3	R3	**R1**	R3	**R2**	R3	R3	R3

3 Tests and Results of Accuracy

At first, 16 users were submitted to the application without the knowledge about the type of command that they should speak to the system. By this procedure, the natural spoken phrases were registered and added to the sub-rule *complemento*. As a high rate of acceptation was noticed, an important question was made: Is the system really recognizing what is spoken by the user? To answer this question, all the words (tokens) really recognized were logged. We could clearly detect incompatibilities between spoken and recognized words and as result of the log analysis we had:

1. The rate of acceptation of all the simple and complex commands was 98%. However just 24.1% really match what was spoken by the user, becoming 85.7% when disregarding the presence of definite articles.
2. The definite articles were recognized in 10.9% of the selected simple commands and from these commands 18.6% were not right. And curiously the rate was 35,3% for selected complex commands and just 6.5% of them were not right.

3. In tests with commands containing numbers from 1 to 32 written as words and in numeral form the recognition was alternated. The recognition in numeral form had the rate of 66.8%. For 34.3% of the numbers we just had the recognition in the numeral form that is what happened with the numbers 7, 14, 19, 23, 24, 25, 26, 28, 29 and 32.
4. The numbers with the highest rates of errors in the recognition was 21, 27 and 31. We noticed the system mistook words with similar sound for numbers like “20 eu” for the number 21. This happened in 70% of the cases in utterances of the number 31, being changed to characters like “trinta ele o”, “trinta aí eu”, “30 aí vou”, “30 aí eu”, “30 aí o”, “30 aí os”, “30 aqui os”, “30 aqui eu”, “30 eu”, “30 em”.

4 Conclusion

In this article we study the behavior of a voice interface system and the implications in the design of grammars to define the voice commands. This study was accomplished using experiments with users, redesigning of a grammar with recursive rules and creating a log to analyze and adjust the grammar. We noticed that the presence of many sub-rules, even with few terminal symbols, demands more computational resources than the opposite. So, the adoption of a small vocabulary in a grammar does not guarantee a low computational cost or accuracy in the recognition.

The use of the redesigned grammar made especially to the application and with good testing brings better recognition accuracy, because it will allow prediction of the next word. This is crucial to the critical systems and decision, where the recognition must be precise. Multi-user coverage without the need of training is a fundamental feature of the voice interfaces of the present days. From this study, we can create a methodology for automatic generation of grammars for interactive applications with proper care about the design of rules. This work intends to helping the coming of an era when interfaces will be more natural to people.

Acknowledgements. We acknowledge CAPES for the financial support and Hueber Candido de Lara and all the other colleagues that helped being part of the tests.

References

1. Burstein, A., Stolzle, A., Brodersen, R.W.: Using Speech Recognition in a Personal Communications System. In: Communications, 1992. ICC 92, Conference record, SUPERCOMM/ICC ’92, IEEE, Los Alamitos (1992)
2. Pfaff, G.E.: User Interface Management Systems, p. 72. Springer, New York (1985)
3. Seneff, S.: TINA: A Natural Language System for Spoken Language Applications. Comput. Linguist. 18, 61–86 (1992)
4. Sun Microsystems Ltd, Java Speech API Programmer’s Guide Version 1.0, [online at], http://java.sun.com/products/javamedia/speech/
5. Vieira, R., Lima, V.L.: Lingüística Computacional: Princípios e Aplicações. In: JAIA – ENIA, 2001, Fortaleza (2001)