AUTOMATIC GENERATION OF VISUAL SCENARIOS FOR SPOKEN CORPORA ACQUISITION

D. Aiello[1], C. Delogu[1], R. De Mori[2], A. Di Carlo[1], M. Nisi[1], S. Tummeacciu[1]

[1] Fondazione Ugo Bordoni, v. B. Castiglione 59, 00142, Roma, Italy
e-mail: demetrio;cristina;adicarlo@fub.it
[2] Université d'Avignon et des Pays de Vaucluse, BP 1228, 84911, Avignon Cedex9
e-mail: renato.demori@lia.univ-avignon.fr

ABSTRACT

The paper describes a system, in JAVA, for written and visual scenario generation to collect speech corpora in the framework of a Tourism Information System. Methods and experimental results are also presented for evaluating the degree of understanding of the proposed scenarios. The corpus generated from visual scenarios appears to be much richer than the one generated from textual descriptions.

1. INTRODUCTION

Corpora of spoken signals are essential to evaluate the parameters of various models used for automatic speech recognition and understanding (ASRU). Corpora are also required for performance evaluation of an ASRU system.

In particular, computer understanding of a spoken message consists in the generation of a data structure which is a conceptual representation of what has been said. This generation is driven by a Knowledge Source (KS) of the many ways in which a concept can be expressed in natural language. Such a KS can be a statistical language model (LM) automatically inferred from the examples of a corpus.

Collecting a corpus is a difficult task. First of all a group of speakers has to be selected and what they are expected to say has to be somehow defined. This paper addresses the second item, a group of speakers having been chosen trying to achieve uniformity along the dimensions of gender, age and regional accent.

If the purpose of the corpus is that of inferring the LM probabilities to be used by an ASRU system, then the application domain and the type of Person-Machine Communication (PMC) are usually defined before planning corpus collection. If, for example, the PMC type is dictation and the domain is the one of a journal, then corpus collection may be based on the dictation of articles already published in the journal.

For applications other than dictation, e.g. for spoken data-base query or for speech-to-speech translation, the corpus should contain spontaneous speech elicited from speakers without presenting them with a text to read.

In fact, *situation* descriptions should be created to act as constraints on what speakers will say by choosing their proper words and sentences. Situations are descriptions of an aspect of the application domain.

The question addressed in this paper is: in what form and what information should be presented to a speaker in order to elicit useful sentences? The analysis described in the following focuses on two types of presentation, namely, textual and visual.

Textual presentations consist in texts describing user situations. Analogously visual presentations consist in displayed scenes showing user situations. Speakers, acting as users, have to suppose to be in one of these situations; they have to say something appropriate to the described situation and consistent with the objectives of the application which has been described before the acquisition sections.

As opposed to the case of newspaper dictation, texts or scenes which have to inspire speakers have to be generated by a procedure that respects the constraints imposed by the application. If such a generation is made by computers, it is convenient to follow a generation paradigm based on a formal method. Such a method should generate a sort of conceptual representation from which *sentence contents* are derived in visual or textual description form. Casual speakers are unlikely to be familiar with formal representation of concepts. For this reason, they are required to express by words their reaction to an intuitive representation of a situation description.

In order to elicit spontaneous sentences for spoken corpora collection, the use of scenarios is crucial. A *scenario* can be defined as a description of an actual task that each speaker, acting as user, has to accomplish using the system. Scenarios should stimulate the subjects to produce a corpus with a large variety of words and language constructs. In general, textual scenarios (TS) are used in corpus acquisition. The limit of these scenarios is that they are likely to influence a speaker in the choice of the words used to express the concepts. So far, there isn't a well-known method for designing scenarios avoiding the linguistic bias introduced by textual scenarios. Some work used table-scenarios [1], or inserted graphic representation in textual scenarios [2].

As the objective is that of producing a rich training corpus with the greatest variability of sentences expressing the same conceptual structure, it is important to check whether the objective is better achieved with textual or visual presentations or a combination of both. A written text

presentation is likely to influence a speaker in the choice of the words used to express the concepts, while visual scenarios (VS) may inspire a greater variety of sentences, even if understanding computer graphics may be more difficult than understanding text.

This paper describes a working system, written in JAVA, for generating sentence contents in textual and visual forms for Traveller Domain tasks of the EUTRANS project for speech-to-speech translation [3].

The focus of this paper is on the generation process. Methods and experimental results for evaluating the produced corpus are also briefly introduced.

2. SITUATION DESCRIPTION GENERATION

It is useful, in practice, to consider scenario generation as a sequence of two processes. The task of the first process is to produce a formal description of the expected spoken message contents, while the second process has to compose a text or a display image corresponding to the description. In the case of images, the second process can be further subdivided into two steps. The first one is a generator of the visual components that should appear in the scene, together with their logical relations. The second one has to assemble the scene elements in such a way that they satisfy the constraints imposed by logical relations and those imposed by geometric consistency. An example of a logical constraint is that a bed in a hotel room should lie on the floor, while an example of geometric constraint is that two physical objects cannot share the same physical space.

A scheme of the proposed generation paradigm is shown in **Figure 1**.

The number of descriptions to be produced is limited by the planned size of the data acquisition effort. Nevertheless, attention has to be paid to cover as much as possible all the aspects made evident by the task analysis. Furthermore, descriptions should stimulate the subjects to contribute to corpus sentence with a large variety of words and language constructs. Redundancies can be tolerated if they do not lead to interpretation errors by the subjects nor introduce excessive cognitive effort for transducing the observations into spoken messages.

Scenario descriptions are conceptual structures; they are made of elementary concept descriptors and their relations. Descriptors and relations are obtained by a generative formalism controlled by processes of constraint generation and satisfaction that ensure coherence in the generated structures. Eventually, these structures should be embedded into a constructor method with which a scenario can be composed in textual or visual form.

Suitable theoretical frameworks for obtaining descriptions are generative grammars and inference systems.

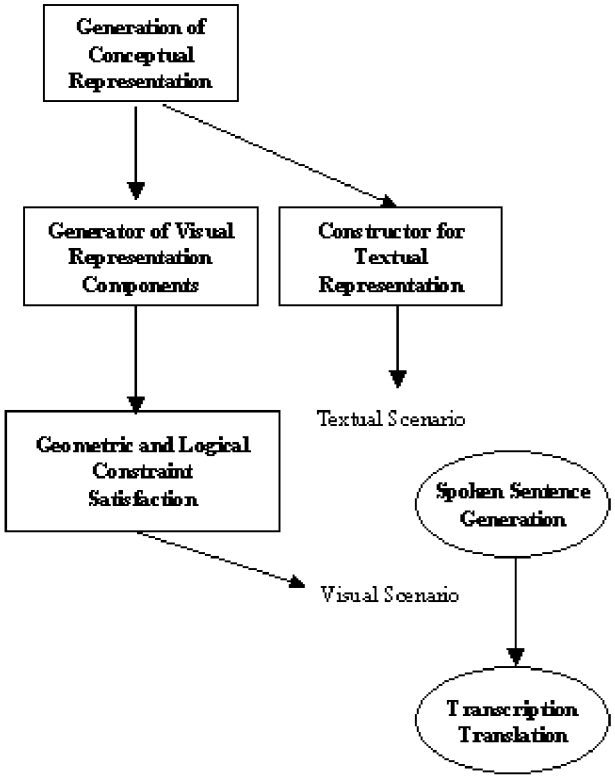

Figure 1: Sentence content generation scheme

Context-free grammars (CFG) are widely used to describe programming languages and are appealing for generating conceptual descriptions represented by functions which can be used to build a text or an image. These grammars deal with elementary concept representations and their composition and can be seen as semantic grammars. It is difficult, if not impossible, to represent dependencies with context-free grammars. Furthermore, context-free grammars describe precedence relations which may not be useful for conceptual representations in which the components in a set can be assembled in many ways. It is thus convenient to avoid unnecessary complexity in the grammar and use it for the generation of sets of constituents to be used in a conceptual representation. Constituent consistency can be verified by augmenting the grammar with constraint generation and satisfaction procedures to be executed with the application of rewriting rules.

There are essentially three types of constraints. A first type consists of constraints expressing domain-independent knowledge, such as the fact that room furniture has to be inside the space of a room. A second type of constraints is domain-dependent such as the fact that a price is usually associated with a hotel room. A third type is that of geometric constraints to be applied when scene components are placed on a display layout.

The first two types of constraints are integrated into a Semantic Augmented Context Free Grammar (SACFG) in which augmentations contain rules for generating constraints and verifying constraint satisfaction. When more than one

rewriting rule can be applied in a derivation, the rule that is effectively applied is randomly selected among the possible candidates.

The generative grammar generates frame data structures.

A frame has a frame header and (attribute, value) pairs. The values of pairs can be nonterminal symbols to be further expanded into frame structures. In some cases, the grammar generates only the attribute part of the pair and lets the human subject use her/his fantasy to decide the values.

The start symbol of the grammar σ represents the class of all possible scenarios.

The first rule is of the type :

$$\sigma \rightarrow \text{FR1} \mid \text{FR2} \mid \text{FR3} \mid .....$$

where FRi are non-terminal symbols from which frame structures are generated. For example, FR1 is rewritten into the following frame structure:

$$\text{FR1} \rightarrow \left| \begin{array}{l} \text{HOTEL\_ACTION} \\ \text{actor FRACTOR} \\ \text{action FRACTION} \\ \text{place FRPLACE} \end{array} \right.$$

The structure for a HOTEL_ACTION is a collection of attributes, *actor*, *action*, *place* and corresponding values which are, in this case, nonterminals generating other frame structures.

FRACTION is a simple non-terminal that can be rewritten into non-terminals generating frames representing a request for information, a cancellation of a reservation, a complaint, a modification.

FRACTOR generates frames about the purpose of the intervention and the other human subjects (himself, wife, friends, etc.) for which the action is performed.

FRPLACE generates frames describing hotels, airports, railway stations etc.

A *derivation* of the grammar is a complete hierarchy of frames which no longer contain non-terminal symbols.

Functions are associated to grammar rules. They provide constraint satisfaction and assemble in a coherent way the components of a situation description.

A method of the JAVA class *Syntax* randomly selects and apply rewriting rules and the associated functions to generate a symbolic representation of a situation.

A natural language description of the scenario is then generated by rules starting from the conceptual structure represented by the result of the derivation.

3. TEXT GENERATION

Every terminal symbol of the grammar is associated with a text *generation function*. In each of them, there are two kinds of statements: first-type statement can directly generate and display one or more phrases in a Natural Language; second-type statements are able to set the display position for phrases generated and displayed by some other *generation functions* recursively called.

4. SCENARIO GENERATION

In order to produce a graphical image, another set of rules is used to generate *instances of JAVA* classes from the specific grammar derivation. Specific constraints can also be obtained by these rules.

Scenarios are generated by the methods of a class *DrawScenario* with the help of a *layout manager*, a classical interface component (JAVA has a number of them) that receives image components and their relations and produces the 2-D image by setting size and position of each image.

This component is still driven by a constraint satisfaction algorithm that is based on geometric constraints.

The following example shows the graphical generation of a situation expressing a taxi request made from a hotel guest to the reception. The conceptual description is represented by the following predicate composed by the functions associated to the grammar rules:

```
Scene (  Actor:    Generic,
         Action:   Service Request ( Object:    Taxi )
         Place:    Hotel Room,
         Hotel:    Hotel (  Name:    Excelsior,
                            City:    Rome ) )
```

Each function is associated with one or more visual instructions. A visual instruction consists of an image and its layout constraints, that allow the *DrawScenario* methods to place, overlap and stretch the basic images in order to compose the final visual representation. In **Figure 2** the set of visual instructions used to build the example situation is graphically represented: each image is connected to its final destination. In **Figure 3** the visual representation is finally shown.

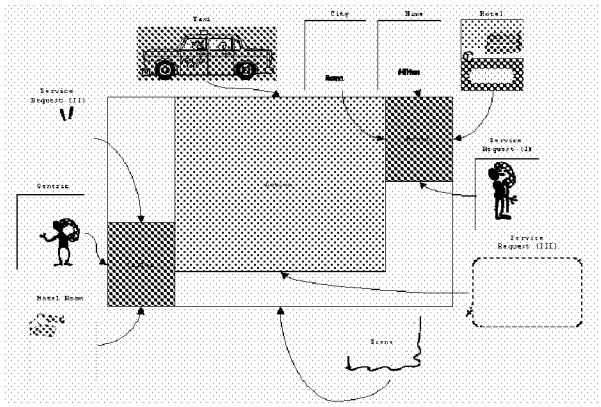

Figure 2: The set of visual instructions used to build the representation of a taxi request.

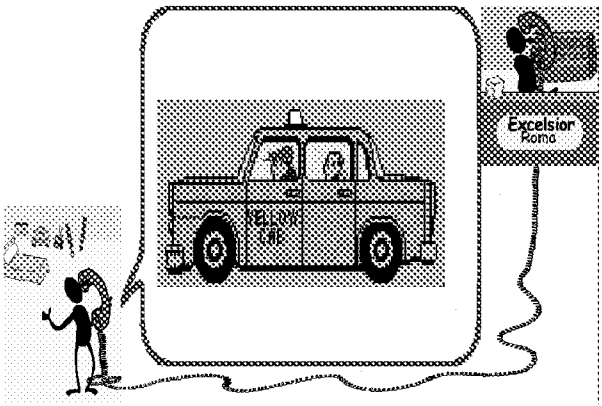

Figure 3: The resulting visual scenario.

5. EVALUATION

With the aim of exploring the possible difference between textual and visual scenarios, an evaluation test has been performed with 100 subjects subdivided into two groups. Textual scenarios were presented to speakers in group-TS while speakers in group-VS were exposed to visual scenarios.

In order to verify the following hypotheses: (i) sentences obtained by VS are more complex than those obtained by TS; and (ii) sentences obtained by VS are more difficult to model than sentences obtained by TS; perplexities were computed using the CMU-Cambridge Statistical Language Modeling Toolkit v2 [4].

A trained set of 400 sentences was used (200 visual and 200 textual) with 8515 words and a vocabulary of 898 type. The test set for the VS group was made of 78 sentences (with 1594 words), while the test set for the TS group was made of 78 sentences (with 2119 words).

Table 1 shows the perplexity computation for both the trigram and the bigram models on the sentences produced starting from visual scenarios (Visual Corpus) and those produced from textual scenarios (Textual Corpus).

The results of scenarios evaluation support the hypothesis that textual scenarios influence speakers in the choice of the words used to express the concepts more than visual scenarios. Furthermore, they also show that the sentences produced by Group-VS have very high lexical differentiation expressed by the higher value of Out-of-Vocabulary (OOV) words.

TRIGRAMS MODEL (a)

	Perplexity	**Out-of-Vocabulary**
Visual Corpus	**36.92**	**120**
Textual Corpus	**20.73**	**77**

BIGRAMS MODEL (b)

	Perplexity	**Out-of-Vocabulary**
Visual Corpus	**41.44**	**120**
Textual Corpus	**26.05**	**77**

Table 1: Perplexity values and out-of-vocabulary words obtained with the trigram model (a) and the bigram model (b).

6. REFERENCES

1. Delogu, C., Di Carlo, A., Sementina, C., Stecconi, S. (1993). A Methodology for Evaluating HumanMachine Spoken Language Interaction. In *Proceedings of Eurospeech'93,Berlin 1993,Vol. 2, pp. 1427--1430.*

2. Dybkjaer, L., Bernsen, N.L., Dybkjaer H. (1995). Scenario Design for Spoken Language Systems Development. In *Proceedings of ESCA Workshop on Spoken Dialogue Systems, Vigso 1995, pp. 93—96.*

3. http://hermes.zeres.de/Eutrans/eutrans.html

4. Clarkson P., Rosenfeld R. (1997) Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proceedings of Eurospeech'97, Rhodes 1997, vol. 5, pp. 2707-2710*