# Interactive Disambguation of Natural Language Input: a Methodology and Two Implementations for French and English

Herve Blanchon

GETA-CLIPS
BP 53, 38041 Grenoble Cedex 9, France
herve.blanchon@imag.fr

## Abstract

As natural language is highly ambiguous even in restricted domains, interactive disambiguation is seen as a necessity for achieving more robust and user-friendly interactive systems, face-to-face translation systems and Dialogue-Based Machine Translation systems. We have proposed a methodology which distinguishes between two parts in a disambiguation module: an engine (language- and application-independent) and a lingware (language- and application-dependent). The engine is, thus, to be reused in the design of any disambiguation module. This paper presents the current state of our work, that is: an engine that has been used to design two interactive disambiguation modules, for French, and English.

## 1 Introduction

### 1.1 Situation

Spoken, written or multimodal natural language is going to be used more and more as the interaction modality between human users and interactive software. Natural language processing techniques do not, and will not, allow a really robust, fault-tolerant and user-friendly utilization of these modalities.

### 1.2 Interest

For the use of natural language to be more robust, fault-tolerant, and user-friendly, we propose to integrate a disambiguation module as a component of every relevant piece of interactive software. The role of such a module is to help the recognizer (for speech) and the analyzer (for text) to produce an unambiguous representation of the user's input corresponding to his intention.

We propose an interactive disambiguation methodology which is based on a clear distinction between two components: a language-independent one (the engine), and a language- and application-dependent one (the lingware).

As the language independent part, the engine should be used in every designed disambiguation module. For each specific application, a specifically designed lingware should be produced as automatically as possible.

### 1.3 Presentation

We will first introduce our basic ideas and proposals. In the second part we will present the current content of the lingware and then the existing implemented engine. The fourth part is dedicated to a brief presentation of our first experiments with the lingware. A first disambiguation module has been designed and implemented, for French input, within a first mockup in the framework of the LIDIA project of Dialogue-Based Machine Translation [Boitet and Blanchon, 1995]. A second one has been designed and implemented, for English input, in the framework of speech-to-speech translation.

## 2 Basic Ideas & Proposals

### 2.1 Software architecture

In the architecture we propose, an interactive disambiguation module is made of two components.

- An engine, which is the core of the module and is language-independent. It will be used in all the disambiguation modules to be developed.
- A lingware, which is language-dependent. It constitutes input data for the engine so as to instantiate a particular disambiguation module.

We are thus aiming for application independence, with the restriction that the linguistic structures produced by the application can be handled in the proposed framework.

### 2.2 Lingware

Ideally, we would like to provide the designer of a disambiguation module with a set of tools allowing him/her, at least, to describe[1]:

- the ambiguities to be solved, even though most of then should be discovered automatically,
- the labeling of the questions to be asked to solve these ambiguities,
- the order in which the ambiguities should be solved, if several are present,

* the modalities to be used to solve each ambiguity,

[1] The items introduced by "-" have already been investigated, the items introduced by "*" have not been investigated yet.

- the modalities to be used to answer the questions about each ambiguity,
- the way questions should be prepared and displayed.

## 2.3 Engine

The engine is then designed to use the lingware to instantiate an interactive disambiguation process. It should provide[1]:

- an ambiguity description language,
- an augmentation tool to increase automatically or semi-automatically the number of recognized and solved ambiguities,
- an ambiguity recognition mechanism to be used to recognize the ambiguities,
- a set of operators to be used to describe the construction of the labeling of the dialogues,
- a mechanism to realize the ordering of the ambiguity recognition process,
- predefined dialogue classes corresponding to the possible modalities,
- a question presentation mechanism,
- a set of question preparation, display and answering strategies.

# 3 The lingware

The structure manipulated in our context is a tree structure called an "mmc structure." This stands for "multisolution, multilevel and concrete." The ambiguities are thus described in terms of tree structures.

## 3.1 Ambiguity description

A type of ambiguity is described as a set of patterns (a beam) (Fig. 1). A pattern describes a tree structure, with variable parts (forests), with constraints on its geometry and labeling.

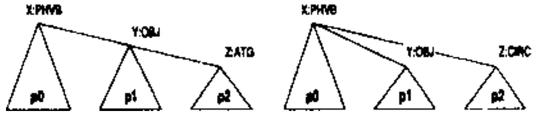


Figure 1. Patterns for a prepositional attachment ambiguity

The previous figure (Fig. 1) shows the graphic description of a beam made of two patterns. In those patterns, the nodes arc X, Y, and Z; the forest variables are P1, P2, and P3. This graphical representation makes explicit the geometrical constraints.

Figure 2 shows the internal representation of the left hand side pattern shown in the previous figure.

```
(defvar *phvbprepatt-tl-1*
 (make-instance 'pattern
  :pattern-name '*phvbprepatt-t11l*
  :pattern-value
    '((?is ?x node-prop-equal-p 'CS 'PHVB)
      (?+ ?p0)
      ((?is ?y node-prop-equal-p 'FS 'OBJ)
       (?+ ?pl)
       ((?is ?z node-prop-equal-p 'FS 'ATG)
        (?+ ?p2))))
 :pattern-method #'item-production-method))
```

Figure 2. The pattern *2phvbadvatt-t1-1*

It is shown that a pattern has a name, a value (its description), and a method.

The node variables are ?x, ?y, ?z; they are constrained on the value of one of their fields. In the first pattern (Fig. 2):

- the value of the field CS of the node ?x must be PHVB,
- the value of the field FS of the node ?y must be OBJ, etc.

The forest variables are ?p0, ?pl, ?p2. They are constrained on their length by the symbol ?+ stipulating them not to be empty.

The item-production-method is described in § 3.2.

A class (or family) of ambiguity is described as a set of beams called a stack. For example, the ambiguities of coordination can be described as a class of ambiguity, and of course, there are several types of ambiguity of coordination.

## 3.2 Ambiguity resolution

A dialogue item production method is associated with each pattern to describe a rephrasing of the part of the sentence recognized by the pattern. This rephrasing is produced using the forest variables instantiated during a successful match. The variables are manipulated with a set of operators.

The following figures (Fig. 3) show the dialogue item production methods associated with the pattern *phvbprepatt-tl-l*(Fig.2).

```
(defmethod item-production-method
  ((pattern-name (eq1 '*phvbprepatt-tl-l*))
   binding)
 (format nil "~A (~A -A)."
      (text (cdr (assoc •?p0 binding)))
      (text (cdr (assoc '?pl binding)))
      (text (cdr (assoc '?p2 binding)))))
```

Figure 3. Item-production-method ((pattern-name (eql '♦phvbprepatt-tl-1*)) binding)

This method (Fig. 3) will produce a string made of the text of pO, the text of p1 and the text of p2 with the texts of p1 and p2 bracketed together.

## 3.3 Ambiguity recognition state

The order in which the different ambiguity classes are to be solved has to be defined. An automaton is used to implement this order. In this automaton, there is one state per ambiguity class defined. The states are organized so as to uphold this order.

The skeleton of those states is the same:

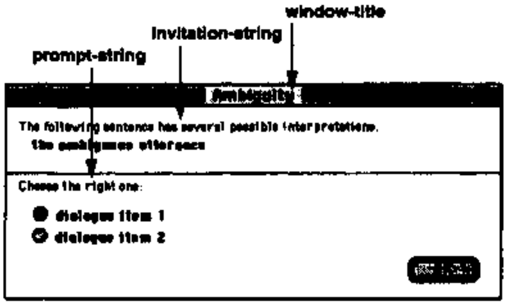
- if one ambiguity of the class of ambiguity to be recognized is actually recognized; then a question has to be prepared and other ambiguities have to be found in the concerned sentence,
- if no ambiguity of the class of ambiguity to be recognized is recognized; then there is transition to the following state in the automaton.

## 3.4 Localized dialogue boxes

In the current implementation the questions to be asked to the user are displayed as dialogue boxes on the screen. These dialogue boxes contain the ambiguous phrase and a set of rephrasing items from which the user will choose the intended one.

The language used to present the question to the user is the language to be disambiguated. Thus there is a need to specify some generic classes that are provided.

For example Figure 4 shows a specification of the generic-textual-clarif-dialog-class textual dialogue box class for English. The invitation-string, the prompt-string, and the window-title are labeled in English. If the question were to be asked in French, the generic-textual-clarif-dialog-class would have to be specified for that.



Ambiguity

The following sentence has several possible interpretations.

the ambiguous utterance

Choose the right one:

● dialogue item 1

○ dialogue item 2

Figure 4. Some dialogue boxes' slots

## 4 The engine

### 4.1 Ambiguity recognition & question construction

#### Pattern & beam matching

*Pattern matching*

The patterns are described with a language derived from the one proposed in [Norvig, 1992]. Pattern matching also inspired by the proposals in [Norvig, 1992] has been implemented.

The result of the pattern matching mechanism is a binding list containing the value, if any, of each variable in the pattern.

*Beam matching*

A sentence S, with s solutions Soli, contains the ambiguity described by the beam B made of b patterns Pj if and only if:

- the number of solutions (s) is strictly greater than the number of patterns (b),
- for each solution Soli there is an unique pattern Pj that matches that solution,
- each pattern Pj matches at least one solution Soli,
- the distance fd between the bindings of the forest variables is null.

#### Operators

Operators are used to describe the dialogue items' construction. They are used by the prepare-question-tree module (cf. Fig. 6) and allowed to perform several operations on the binding of the variables. Three families of operators are defined to perform: selective projection, access to the lexical database, and formatting operations. For a decribed description of the operators refer to [Blanchon, 1995],

#### Disambiguation question class

The disambiguation process produces a question-tree made of disambiguation questions. Those questions are to be asked to the user according to the modalities to be used.

In the current implementation, the questions are displayed as dialogue boxes on the screen. Each disambiguation question is an instance of the predefined class clarification-question-class.

### 4.2 Disambiguation automaton

Each state of the automaton is defined as a CLOS method. There are basically three kinds of states:

- an entry point, that is, the first state of the automaton. It is called automaton-scheduler and provided by the engine,
- meta-ambiguity recognition states provided by the engine, and,
- ambiguity class recognition states provided by the lingware.

#### Automaton entry point

The method automaton-scheduler is the entry point of the disambiguation automaton. If there is no ambiguity to be solved, this state is also the exit point of the automaton. When there are no more questions to be prepared, an empty question is produced which is a leaf of the question tree .

### Meta-ambiguity recognition states

A meta-ambiguity recognition state is a predicative state used as a branching state in the automaton. So far we have proposed three ambiguity meta-classes. These classes are called: lexical-ambiguity, geometrical-ambiguity and labeling-ambiguity. They are refined by the designer of the lingware into several designer-defined classes of ambiguity.

### Scheduling

Thus, a disambiguation automaton should be shaped as shown in Figure 5.

The lexical ambiguities are to be solved first, then the geometrical ones, and finally the labeling ones. This strategy is guided by the following principles:

1 first, find the right simple phrases (i.e. solve the lexical ambiguities),
2 second, find the construction of the verbs (i.e. solve the labeling and some of the geometry ambiguities),
3 third, find the structure of the dependents of the verbs (i.e. solve the geometry ambiguities which did not fall in the previous case),
4 last, find the word senses.

Those criteria seem reasonable and natural. Moreover, the order of the kinds of questions will not be changed to improve the usability of the system.

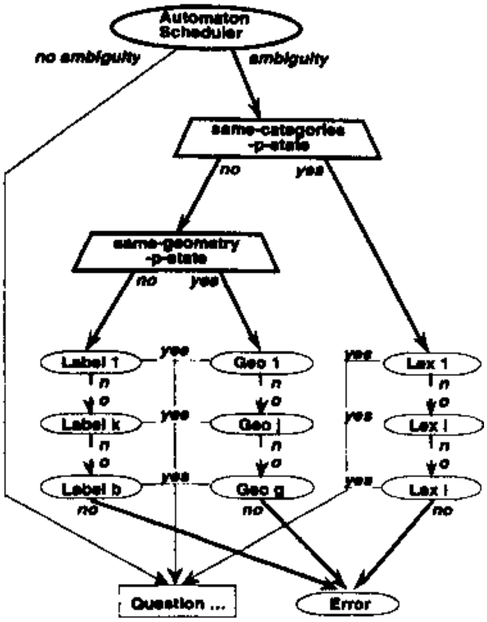


Figure 5. General organization of a disambiguation automaton

## 4.3 Question construction & presentation

### Question construction

Once an ambiguity has been recognized, a question is produced. The dialogue item construction method associated with the recognized patterns are applied using the value of each of the variables used to describe them.

### Question tree construction

The construction of the question tree is a loop in a disambiguation automaton organized by the prepare-question-tree method (Fig. 6).

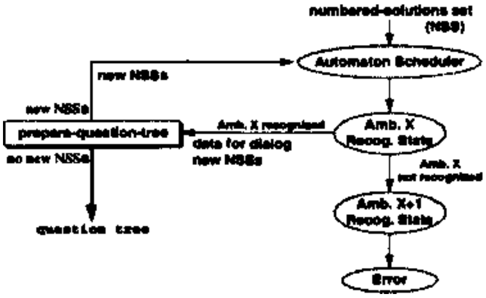


Figure 6. The question-tree construction

When a question tree has been built, it is presented to the user by the question-tree-presentation method.

### Question tree presentation

For a given ambiguous utterance, the disambiguation automaton produces a question tree. The question tree is covered by the question-tree-prcsentation function until no more questions arc to be asked. The method ask-question proposes the question to the user.

## 5 Implementation

So far we have developed two disambiguation modules; one for French and one for English. They are briefly described here.

### 5.1 A French input disambiguation module

#### Context

The module for French has been developed at the GETA laboratory in the framework of the LIDIA project of Dialogue-Based Machine Translation [Boitet and Blanchon, 1995]. It is currently made of 13 beams.

#### Corpus

For the LIDIA-1 mockup we constructed a corpus of ambiguous sentences. Those sentences were chosen from the literature about ambiguity in French [Fuchs, 1987, Vauquois and Nedobejkine, 1977]. The sentences were selected according to their underlying linguistic structure. The wording was not considered.

#### Classification & example

We have defined the following ambiguities described by means of beams.

* Lexical Ambiguity

*Verbal coordination*

- Il atteint la grange et la ferme[2].

* Geometrical ambiguity

*Argument structure of the verb*

- Il parle depuis l'ecole de cuisine[3].
- Le capitaine a rapporte un vase de Chine[4].

*Noun coordination*

- On etudie l'ecolution de la structure de la bourse et des investissements[5].

*Adjective coordination*

- Il prends des cahiers et des classeurs noirs[6].

*Subordination*

- Elle epouse un professeur de droit anglais[7].

#### Dialogues

Here is a set of examples of the produced dialogues.

[2] He reaches the barn and the farm *or,* He reaches the barn and closes it. *

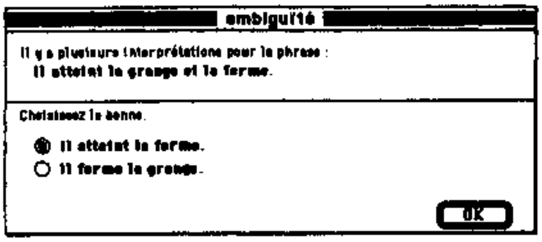
[3] He speaks from the school, about cooking *or.* He speaks from the cooking school.

[4] The captain brought a vase from China *or,* The captain brought a Chinese vase.

[5] We are studying the evolution of the structure of the investments *or,* We are studying the investments.

[6] Black files *or,* Black files and black notebooks.

[7] An English professor *or,* A professor of English law.



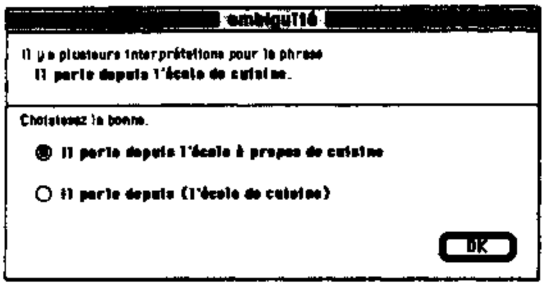
ambiguïté

Il y a plusieurs interprétations pour la phrase :
Il atteint la grange et la ferme.

Choisissez la bonne.

● Il atteint la ferme.
○ Il ferme la grange.

Figure 7. Dialogue for a verbal coordination ambiguity[8]



ambiguïté

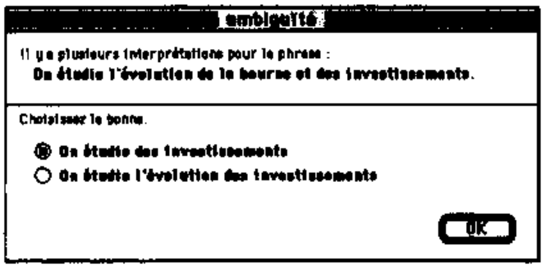
Il y a plusieurs interprétations pour la phrase
Il parle depuis l'école de cuisine.

Choisissez la bonne.

● Il parle depuis l'école à propos de cuisine

○ Il parle depuis (l'école de cuisine)

Figure 8. Dialogue for an ambiguity of argument structure of the verb, type 1 [9]



ambiguïté

Il y a plusieurs interprétations pour la phrase :
On étudie l'évolution de la bourse et des investissements.

Choisissez la bonne.

● On étudie des investissements
○ On étudie l'évolution des investissements

Figure 9. Dialogue for a noun coordination type 1 ambiguity[10]

## 5.2 An English input disambiguation module

### Context

The module for English has been developed in the framework of Speech Translation [Blanchon, 1995]. It is currently made of 10 beams.

### Corpora

The ambiguities which make up the first corpus upon which the clarification mechanism was based were taken from a data base of spontaneous speech. The conversations, between native speakers of American English, were recorded during an experiment conducted in the Environment for MultiModal Interaction (EMMI) [Loken-Kim, et al., 1993], and took place via both telephone and multimedia communication contexts [Fais and Loken-Kim, 1994].

A description of the ambiguities to be solved is given in [Fais and Blanchon, 1996]

### Classification & examples

So far we have defined the following ambiguities described by means of beams,

* Lexical Ambiguity

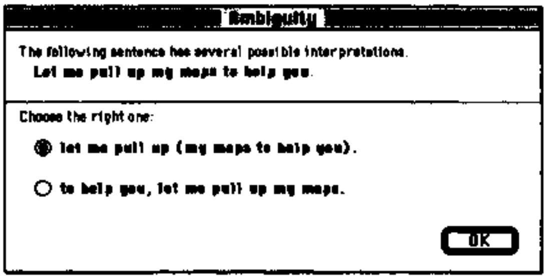
- This is an English speaking agent.

* Geometrical ambiguity

- Let me pull out my maps to help you.
- Where can I catch a taxi from Kyoto station.
- You are going to the International Conference Center.
- I will show you where you are located right now.
- You can pay for it right on the bus
- You can tell him that you are going to the international conference center and it should be a twenty minute ride.

### Dialogues

Here is a set of examples of the produced dialogues.



Ambiguity

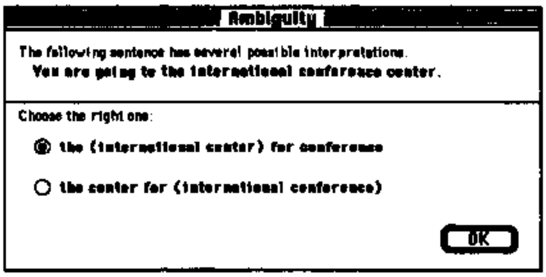
The following sentence has several possible interpretations.
Let me pull up my maps to help you.

Choose the right one:

● let me pull up (my maps to help you).

○ to help you, let me pull up my maps.

Figure 10. Dialogue for "Let me pull up my maps to help you.'



Ambiguity

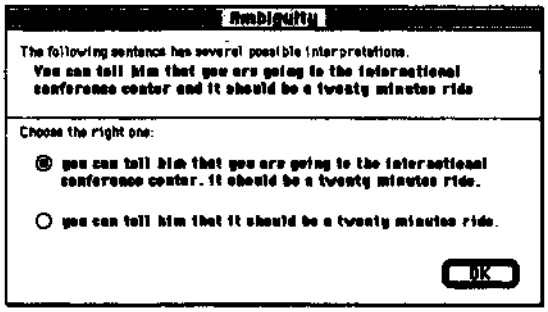
The following sentence has several possible interpretations.
You are going to the international conference center.

Choose the right one:

● the (international center) for conference

○ the center for (international conference)

Figure 11 . Dialogue for "You are going to the international conference center."



Ambiguity

The following sentence has several possible interpretations.
You can tell him that you are going to the international conference center and it should be a twenty minutes ride

Choose the right one:

● you can tell him that you are going to the international conference center. it should be a twenty minutes ride.

○ you can tell him that it should be a twenty minutes ride.

Figure 12. Dialogue for "You can tell him that..."

[8] He reaches the farm, He closes the barn.

[9] He speaks from the school about cooking, He speaks from (the cooking school).

[10] The proposed interpretations are: We study the investments, We study the evolution of the investments.

# 6 Conclusion & perspective

## 6.1 On the methodology

The methodology we proposed should allow the development of customized disambiguation modules that can be easily improved incrementally.

The customizability comes from the clear separation between the lingware and the engine. In this framework, different disambiguation modules can be produced for one or several different languages and kinds of input.

The description of the linguistic data can be augmented incrementally as the design and the use of a disambiguation module progress.

Certainly, we do not claim that any given module will cover all the ambiguities found in natural language. On the other hand, we claim that a given module for a given application can incrementally reach a broad coverage for the application it has been designed to be integrated into

## 6.2 On the evaluation

We think that, whenever a system uses a natural language analysis module, the evaluation criterion must not only be the task completion time. More important is the user satisfaction. That is why we feel that it is very important to study the design of the clarification sessions, and moreover, the design of the questions to be asked. We are aware that this kind of study is energy- and time-consuming but it has to be done to build real-scale usable systems.

Thus, we have proposed to run experiments to study, before other questions, the understandability of the proposed disambiguation dialogues. We have already run two experiments on that topic. The results of these two experiments are also described in [Blanchon and Fais, 1996].

We strongly hope that it will be possible to carry on this work on understandability and assessment.

## 6.3 Technical perspectives

There is of course a lot of work to be carried out to reach our goal. Here are the most challenging ones:

The (semi-) automatic lingware augmentation. That is:

- the learning of new patterns, and beams [Frey, 1995],
- the construction of the dialogue item production methods to be associated with the new patterns),
- the updating of the disambiguation automaton.

The manipulated data structures may be weighted so that the module can adapt itself to the user, and be tunable.

We do also think that it will be necessary to provide an interactive environment to enable the design of a disambiguation module.

The handling and the use of several modalities also have to be studied.

# 7 Related works

Other people have been working on interactive disambiguation, we would recommend [Ben-Ari, et al., 1988, Goodman and Nirenburg, 1991, Maruyama, et al., 1990].

# References

[Ben-Ari, et al., 1988] Ben-Ari, D., Berry, D. m. and Rimon, M. Translational Ambiguity Rephrased, in Proceedings of Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language. (Carnegie-Mellon University, Pittsburgh, 12-14 juin 1988).

[Blanchon, 1995] Blanchon, H. An Interactive Disambiguation Module for English Natural Language Utterances, in Proceedings of NLPRS'95. (Seoul, Dec 4-7, 1995), vol. 2/2 : 550-555,

[Blanchon and Fais, 1996] Blanchon, H. and Fais, L. How to ask Users About What they Mean: Two Experiments & Results, in Proceedings of MIDDIM'96. (Le col de porte, Isere, France, 12-14 AoOt 1996), vol 1/1 : 238-259.

[Boitet and Blanchon, 1995] Boitet, C. and Blanchon, H Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mockup. Machine Translation. 9, 2, 99-132,

[Fais and Blanchon, 1996] Fais, L. and Blanchon, H. Ambiguities in Task-oriented Dialogues. in Proceedings of MIDDIM'96. (Le col de porte, Isere, France, 12-14 Aout 1996), vol. 1/1 : 263-275.

[Fais and Loken-Kim, 1994] Fais, L. and Loken-Kim, K.-h. Effects of communicative mode on spontaneous English speech. Rap. Institute of Electronics, Information and Communication Engineers. Technical Report. n° NLC94-22. Oct. 94. Pages.

[Frey, 1995] Frey, V. De la reconnaissance a la decouverte de "patrons d'ambiguiteV en TAFD. Rap. University Josephe Fourier (MATIS - Archamps). Rapport de DEA. 24 juin 1996. 67 Pages.

[Fuchs, 1987] Fuchs, C. L'ambiguite' et la paraphrase en linguistique. in L'ambiguite' et la paraphrase. Centre de Publication de l'Universit6 de Caen. Caen. 1987.

[Goodman and Nirenburg, 1991] Goodman, K. and Nirenburg, S. (ed.). The KBMT Project: A case study in knowledge-based machine translation. Morgan Kaufmann. San Mateo, California. 1991.

[Loken-Kim, et al., 1993] Loken-Kim, K.-H., Yato, F., Kurihara, K., Fais, L. and Furukawa, R. EMMI-ATR environment for multi-modal interactions. Rap. ATR-ITL. Technical Report. n° TR-IT-0018. Sept 30, 1993. 8 Pages.

[Maruyama, et al., 1990] Maruyama, H., Watanabe, H. and Ogino, S. An Interactive Japanese Parser for Machine Translation, in Proceedings of Coling-90. (Helsinki, 20-25 aout 1990), vol. 2/3 : 257-262.

[Norvig, 1992] Norvig, P. Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp. Morgan Kaufmann Publishers. San Mateo, California. 1992.

[Vauquois and Nedobejkine, 1977] Vauquois, B. and Nelobejkine, N. Ambiguites de la langue ecrite. Rap. Action Thenatique Programmed de Linguistique Gen6rale, Table ronde organised au CNRS. 11 Janvier 1977. 8 Pages.

# NATURAL LANGUAGE PROCESSING AND GRAPHICAL PRESENTATION

## Graphics