TR-IT-0073

Pattern-based approach to interactive disambiguation: first definition and experimentation

Hervé BLANCHON

1994.9.8

The aim of this document is to have the reader acquire a clear view of the pattern-based interactive disambiguation process developed.

The kernel of the disambiguation process has two main entries: the data produced by an analyser and some sets of patterns describing the ambiguities to be recognized and solved. The kernel is thus language independent, which means that disambiguation processes can be developed for other languages using the methodology we used to develop an interactive disambiguation process for French input.

The first part of the document gives a brief description of the LIDIA-1.0 mockup to situate the disambiguation process.

The second part of the document is an in-depth study of the design of the disambiguation process. This part is an important update and correction of [Blanchon 1994a] which is released in French. This work has been done at ATR.

The Appendix A gives a linguistic classification of the ambiguities we have to solve in the field of Natural Languge Processing.

The Appendix B gives a description of the structure produced by the analyzer used in the framework of the development of the LIDIA-1.0 mockup.

The Appendix C gives the list and the meaning of the labels produced by the analyzer which are relevant to the disambiguation process.

ATR – Interpreting Telecommunications Research Laboratories
2–2 Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan

Content

Figures

CHAPTER 5 IDEAS INVOLVED IN THE DESIGN OF THE DISAMBIGUATION PROCESS

CHAPTER 6 REALIZATION & RESULTS

APPENDIX B THE STRUCTURE PRODUCED BY THE ANALYZER

Introduction

Situation, motivation

In the framework of the LIDIA project (Large Internationalization of Documents through Interaction with the Authors) I organized the realization of a first mockup LIDIA-1.0 demonstrated on the 20 of January 1994. During this work, I have been in charge of imagining, specifying and implementing an interactive disambiguation process. This disambiguation process has been designed to solve some of the ambiguities represented in a multisolution, multilevel and concrete tree structure produced during the analysis of French textual input. The mmc-structure is described in the Appendix B, and the labels used in this structure are described in Appendix C.

One of the motivations of this document is to give a detailed account, in English, of the designed disambiguation process. For ATR, this report would be a reference for future work on interactive disambiguation based on the proposed methodology.

Interest

The aim of this document is to have the reader acquire a clear view of the pattern-based interactive disambiguation process developed.

The kernel of the disambiguation process has two main entries: the data produced by an analyser and some sets of patterns describing the ambiguities to be recognized and solved. The kernel is thus language independent, which means that disambiguation processes can be developed for other languages using the methodology we used to develop an interactive disambiguation process for French input.

The next step of this work is to describe and implement an interactive disambiguation process for English input.

Organisation of the document

The first part of the document gives a brief description of the LIDIA-1.0 mockup to situate the disambiguation process.

The second part of the document is an in-depth study of the design of the disambiguation process. This part is an important update and correction of [Blanchon 1994a] which is released in French. This work has been done at ATR.

The Appendix A gives a linguistic classification of the ambiguities we have to solve in the field of Natural Languge Processing.

The Appendix B gives a description of the structure produced by the analyzer used in the framework of the development of the LIDIA-1.0 mockup.

The Appendix C gives the list and the meaning of the labels produced by the analyzer which are relevant to the disambiguation process.

Part I

Overview of LIDIA-1.0: a first mockup towards DBMT for monolingual authors

Introduction

Dialogue-Based Machine Translation (DBMT) is a new paradigm for translation situations where other approaches, such as the Linguistic-Based (LBMT) and the Knowledge-Based (KBMT) approaches, are not adequate. In DBMT, although the linguistic knowledge sources are still crucial, and extralinguistic knowledge might be used if available, emphasis is on indirect pre-editing through a negotiation and a clarification dialogue with the author in order to get high quality translations without revision. Authors are distinguished from "spontaneous" writers or speakers by the fact that they want to produce a "clean" final message and may be willing to enter into such dialogues.

In the first phase the typical translational situation considered is the production of multilingual technical documentation in the form of HyperCard stacks. Notable points in the linguistic design include multilevel transfer with interlingual acceptions, properties and relations, the "guided language" approach (typed textual fragments and lexical preferences), and a TEI-inspired representation of texts and structures. The current mockup, LIDIA-1.0, demonstrates the idea on a HyperCard stack, presenting short ambiguous French sentences in context. This stack is translated into three stacks, German, Russian and English. Some aspects are presented in more detail, in particular the user interface and the object-oriented design. The production of disambiguation dialogues is the subject of the second part of the document. Although this mockup does not implement all features of the general design, because a complete implementation would have called for considerably more human resources than were available, we feel it demonstrates the potential of the approach and is a first step towards a usable prototype, where the linguistic engineering aspects and the reactions of real users could be studied.

For more information on the LIDIA project and the LIDIA-1.0 mockup the reader can refere to:

- [Boitet 1989 ; Boitet 1990b ; Boitet & Blanchon 1993 ; Boitet & Blanchon 1994] for the concept of DBMT,
- [Blanchon 1990 ; Blanchon 1994a ; Blanchon 1994b] for the general design of the mockup,
- [Blanchon 1991 ; Blanchon 1992 ; Blanchon 1994a ; Blanchon 1994b] for the general design of the disambiguation process.

Chapter 1

Overview of the LIDIA-1.0 design

1.1. Choosen situation

In the first phase of the LIDIA project, we are considering a particular situation, where a monolingual French engineer is supposed to create technical documentation, in the form of a HyperCard stack, on a middle-range Macintosh, and to help the system translate it into English, German and Russian. Our architecture is distributed: author's workstation on a Macintosh and MT server on a mini—IBM-4361 running the Ariane-G5 environment [Wehrli 1990]..

1.2. Why HyperCard?

The choice of HyperCard is motivated by the fact that hypertexts are becoming popular supports for technical documentation, and on the assumption that writers will more readily agree to participate in a dialogue if the tool they are using is very interactive than if they use a more classical text processor. Finally, there are some linguistic advantages. First, the textual parts are clearly isolated, and not cluttered with images, formulas, tabs, markups, etc. Second, the textual parts may be typed, thus greatly facilitating analysis. Examination of existing stacks suggest two levels of typing: *utterance styles* in the case of short textual fragments (sentences or phrases), and *text genres* (such as explanation, advice, commented program...) for longer textual fragments.

HyperCard documents are interactive "hypertexts", called "stacks", through which users can navigate. A stack is a collection of cards.

HyperCard interacts with the user by displaying one card at a time. A card has its own buttons and fields, and a background which in turn has buttons and fields. Buttons are "hot spots", which trigger actions if "clicked". Fields contain editable text.

A background may be shared by several cards and a stack can have several

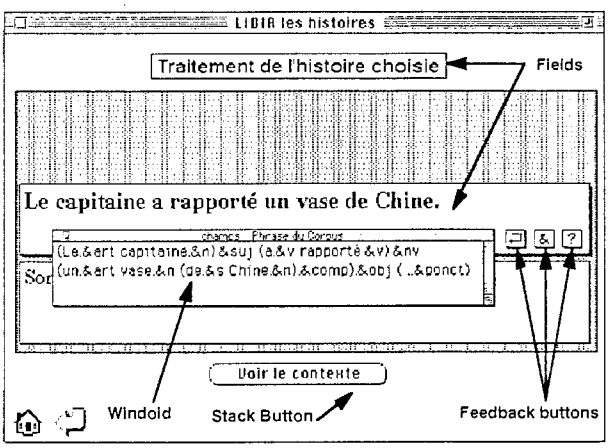


Figure 1.1: A HyperCard card and its objects

backgrounds. A card overlays its background (both have the same size). Pictures may be drawn on cards and backgrounds.

HyperCard offers tools for incrementally and interactively constructing stacks.

1.3. Organization of the automatic and interactive processes

The main processes are illustrated in figure 1.2 below. Many other organizations would be possible. We have settled on this one, which is relatively simple, for practical reasons only.

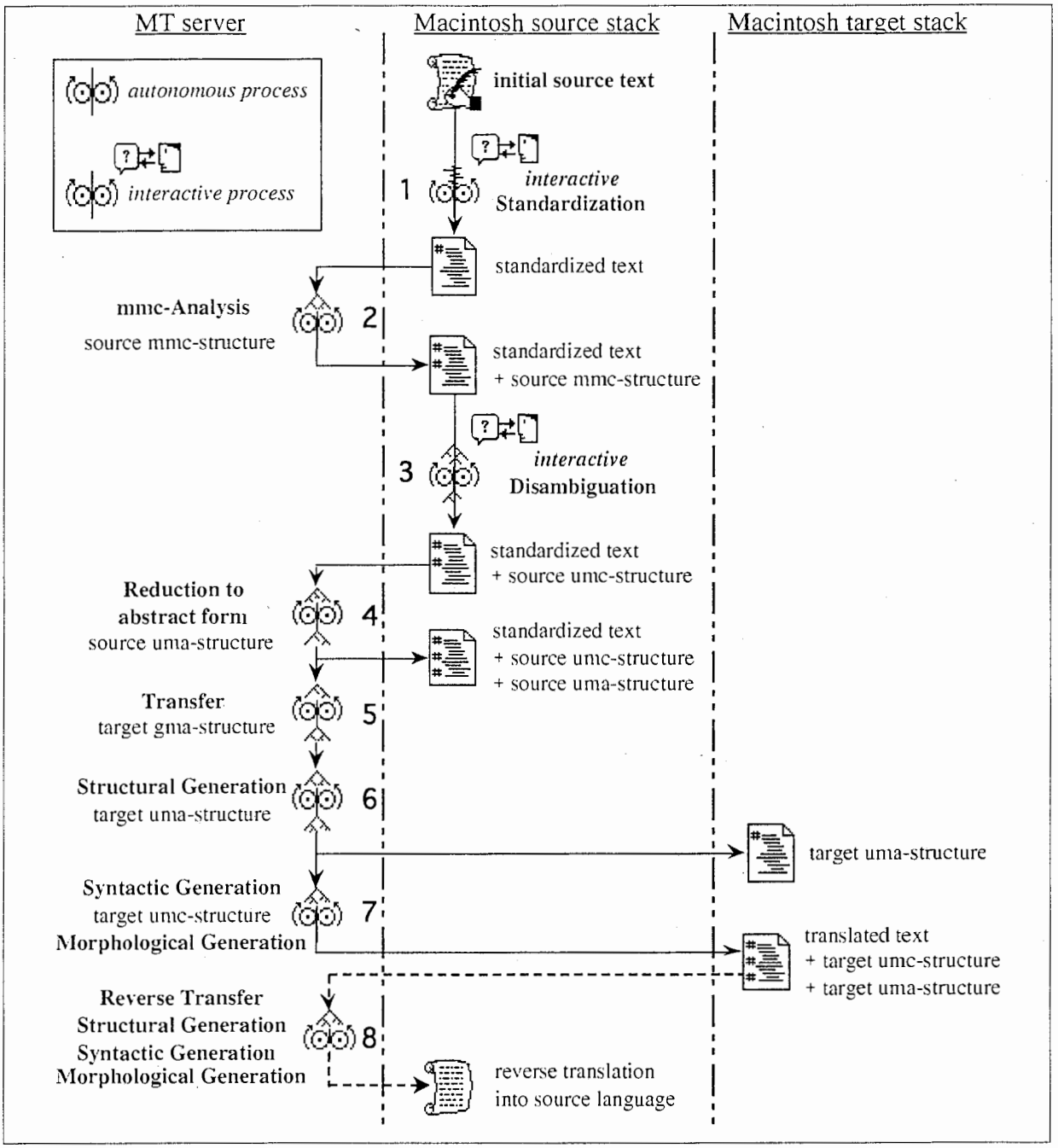


Figure 1.2: General organization of the translation process in LIDIA-1

1. The process of "standardization" aims at helping the automatic translation process, thereby reducing the required interaction. After standardization,

- all fields and buttons should be assigned a text type[1], in order to control stylistic checking and later on to help analysis *(text categorization module);*
- the texts should be well spelled, and conform to the stylistic parameters associated with the type of their containers *(grammar & style checkers);*
- fixed phrases behaving in special ways (such as the menu item Hide Balloons in "Hide Balloons turns Balloon help off") should be marked (the user should assist the *special fixed phrases module* in the construction of the list of fixed phrases used in the stack)[2];
- terminological preferences (e.g., between "plane", "aeroplane", and "aircraft") should be enforced by the *lexical preference module,* whether they loosely reflect local writing habits, or are mandatory because of the domain itself, or because of some normalization.

2. The standardized text is then analyzed on the translation server. The internal representation of the *source mmc-structure* (Multisolution, Multilevel and Concrete) produced is then transformed into an external form directly readable by Lisp and sent back to the Macintosh.

3. The source mmc-structure is used to produce the disambiguation dialogue on the Macintosh. After disambiguation, it becomes an unambiguous *source umc-structure* (Unisolution, Multilevel and Concrete) corresponding to the analysis chosen by the author[3].

4. The source umc-structure is then abstracted, or "reduced" to a *source uma-structure* (Unisolution, Multilevel and Abstract).

5. The system then produces the *target gma-structures* (Generating, Multilevel and Abstract), using adequate transfer components. A gma-structure is in a way more "general" and "generative" than a uma-structure, because its surface-oriented levels (syntactic functions, syntagmatic categories...) may be empty, and if not are just preferences indicated by the transfer.

6. Structural generation produces a *target uma-structure* homogenous with what would be the result of analyzing (and disambiguating) the target text to be generated. It consists of choosing the paraphrase to be generated by computing the surface levels and a first approximation of the word order from the deeper levels (logical relations, semantic relations, semantic features, etc.).

7. The translation process ends with syntactic generation and morphological generation. When all objects of the source document are translated, we get image stacks in the target language(s).

8. The target uma-structures may be used to help the (monolingual) user control the translation by performing a reverse translation into the source language.

---

[1] In the case of "incomplete" texts, where for example the subject of the first sentence is contained in another field (as in tables containing command names and their explanations), this module also asks how to construct the complete text.

[2] The two preceding modules can work directly with the text as written by the author. From here on, however, the system works on a transcription contained in the "shadow record" associated with the card, as well as with intermediate forms of processing. This forces us to lock the original textual field (unless the author decides to change it and is willing to start all over again).

[3] "Concrete" means that the original text can be read from the structure in a direct way (by inorder traversal of the leaves in constituent structures or of all nodes in dependency structures). The nodes and/or edges of the structure may contain "surface" information as well as "deep" information (predicate/argument organization, semantic relations...). In "abstract" structures, negations and auxiliaries may have been suppressed as nodes and represented in decorations, elided elements may have been restored, ordering may have been normalized, etc.

Chapter 2

LIDIA-1.0: choices, limitations, and demo

2.1. Choices and limitations specific to LIDIA-1.0

2.1.1. Translatability constraints on HyperCard stacks

How the user interacts with HyperCard is defined by a set of preferences. We have added a new preference item (a check box) to start or stop LIDIA.The structure of the stack must be fixed before invoking LIDIA, which means that LIDIA-1.0 can not track the creation or deletion of objects.

We have also put some restrictions on the structure of "translatable" stacks, which ensure that we don't need to translate the scripts[4], but only the names of the buttons and the textual contents of the fields[5]. Hence, the scripts should be written in a language-independent way. This implies that:

- messages should never be contained within scripts, but always retrieved from normal fields[6], which will be made invisible in the final version of the stack;
- button references should not be names, but numbers, invariant in translation;
- text drawn within pictures will not be translated;
- any customized version of the menu bar will have to be translated by hand.

2.1.2. Standardization and disambiguation dialogues

In this mockup, no standardization tools such as spell-checker, style-checker, and text categorizer have yet been included, although some preliminary experiments have been done with "Le Correcteur 101" by Machina Sapiens. There is only a very primitive typing of textual elements: Normal, Title, and Don't_Translate.

---

[4] Every HyperCard object has a (possibly empty) *script*, written in the HyperTalk programming language. A script is a collection of *handlers*, each of the form "on <message> do <sequence of HyperTalk statements> end". A handler is *invoked* when its <message> (such as a mouse click) is received by the object whose script contains it.

[5] The assumption is that a stack is fully translated into as many stacks as there are target languages. Another possibility would be to make the stack multilingual by creating a copy of each text container for each language.

[6] However, they may contain variables: "Please locate file &1" should certainly be translatable.

Dialogues are conducted only through the screen. We hope to experiment with the introduction of speech synthesis in disambiguation dialogues in the future. In order not to overload the users with new things to learn, we have also preferred to stick with menu-driven dialogues. For example, we have considered using graphic manipulation of structures (which might be represented as graphs, or as embedded boxes) for structural disambiguation, but potential users found it more difficult than to choose between textual rephrasings (examples below).

2.1.3. Linguistic coverage

The lexical coverage of the mockup is rather small: 134 French lemmas, corresponding to 526 acceptions, 304 English lemmas, 370 German lemmas, and 394 Russian lemmas. However, the entries have not been simplified in any ad hoc way. There are in fact a lot more lemmas for each of these languages under Ariane-G5 (≈30,000 for Russian, 10,000 for French and English, 5,000 for German), but they are not yet coded for the acceptions.

The grammatical coverage is medium. Not all constructs are treated, but a lot more are treated than appear in the demo corpus and in the test corpus. Also, grammars are not ad hoc. The test corpus comprises textual fragments (phrases, sentences) taken from some experimental HyperCard stacks. The demo corpus is quite small (10 stories, each consisting of 2 or 3 sentences).

2.2. A demonstration stack

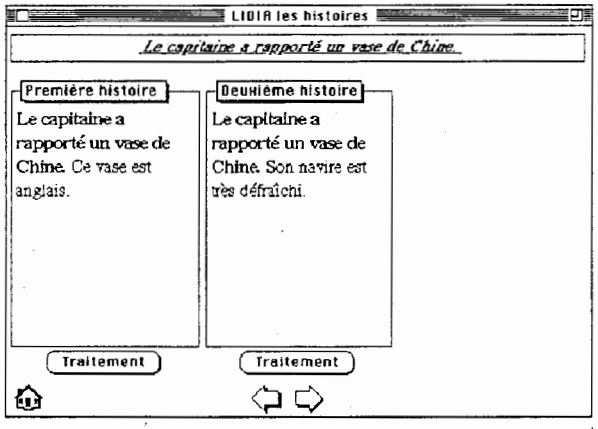


Figure 2.1: A story card[7]

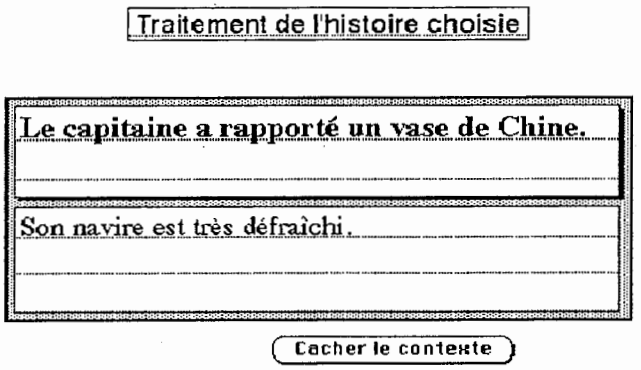


Figure 2.2: A sentence card

The demonstration stack, 'LIDIA les histoires', has two types of cards, story cards and sentence cards. The idea is to show that interactive disambiguation is both necessary and possible, by putting some ambiguities in contexts where no expert system could reliably solve them, and they would not carry over to the target languages.

A story card is a collection of two or three stories having an ambiguous sentence in common. The author is supposed to solve the ambiguities through his understanding of the stories. Each story is presented in a sentence card, where the context of the ambiguous sentence may be shown or hidden.
The DBMT process is activated by asking for the translation of any field of a sentence card. Note that the user is never interrupted by a question. Objects show they are waiting for answers, and the user decides which question to answer and when.

---

[7] The story of the left can be translated as: '**From China, the captain has brought back a vase.** This vase is English'. The second story can be translated as: '**The captain has brought back a Chinese vase.** His boat is soiled.'

2.3. Example session with LIDIA-1.0

The author checks the LIDIA HyperCard preference item. The Macintosh starts to periodically connect to the MT server, in much the same way as a mail utility. The user may continue to navigate in the stack and edit fields.

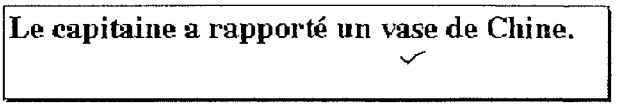


Figure 2.3: Selection of an object

When s/he decides that some objects (fields, buttons) are ready to be translated, s/he chooses the translation tool in the LIDIA palette, selects the objects with LIDIA's selection tool ✓ (fig. 2.3), and continues to work while translation-related processes execute.

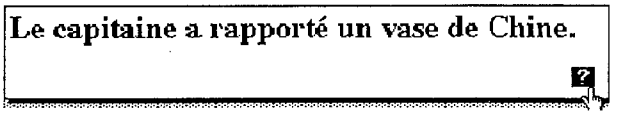


Figure 2.4: Status watcher for a field

The translation status of any object can be monitored through its a *status watcher,* which is automatically created when the object is selected. When clicking on it (fig. 2.4), a windoid or pop-up window appears (fig. 2.5).

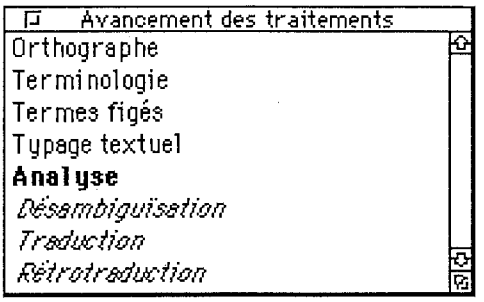


Figure 2.5: Status watcher windoid

The task in progress is displayed in bold, the previous ones in plain, and the following ones in italic. Thus, in figure 2.5, the system is currently analyzing the text fragment.

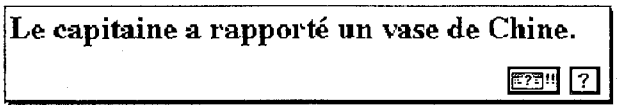


Figure 2.6: Questions on an object are waiting

When intervention by the user is required, LIDIA sends out a (parametrizable) signal, exactly like background utilities such as PrintMonitor or Mail. The user is free to interact at that point or later.

To interact with an object (as small as a button or as big as the entire stack), one simply double-clicks on its status watcher and selects the appropriate item in a pull-down menu. The interaction mode can be exited at any time.

After analysis, the sentence may have to be disambiguated. A new item is added to the menu Message and a new button appears over the concerned object as in figure 2.6. The user can choose to interact at once or later.

Suppose the user clicks on the button. A first question appears (fig. 2.7). In the context of this story, the user should choose to attach 'de Chine' to 'vase' (Chinese vase). A second dialogue appears (fig. 2.8) to ask about the word sense of 'capitaine'.

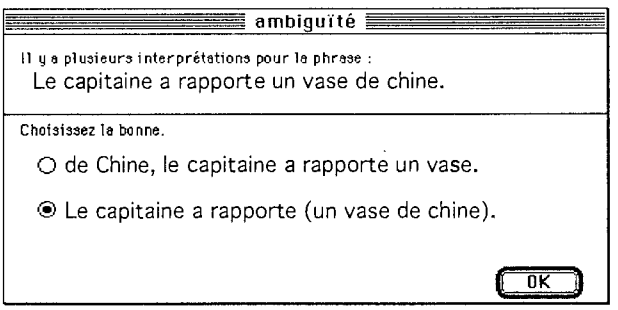


Figure 2.7: Attachement problem (story 2)

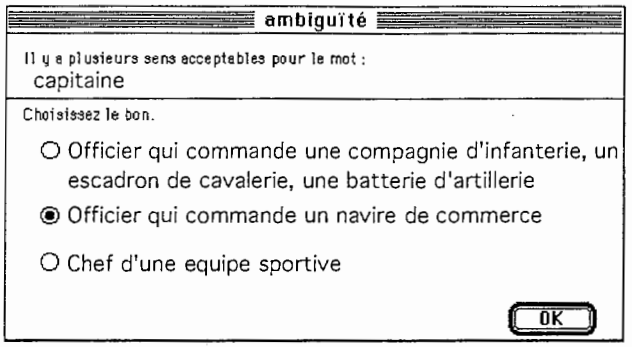


Figure 2.8: Word sense disambiguation (story 2)

These word senses are retrieved from Parax, our HyperCard mockup of a multilingual lexical database [Sérasset 1994].

That is all for this sentence. To see the annotated form of the text, one selects the appropriate item in the LIDIA menu. In LIDIA-1.0, we produce only one view, containing the syntactic class of each occurrence and the syntactic function of each phrase (see fig. 2.9).

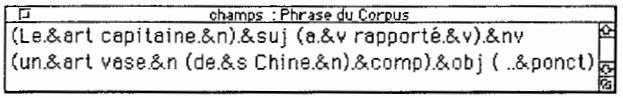


Figure 2.9: View of the annotated form

To check the translation produced in a target language, the user can ask for the "reverse translation", produced from the abstract structure (uma-structure) of the target text. The example concerns the second interpretation of the example (fig. 2.10).

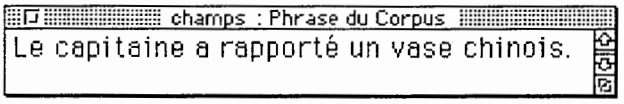


Figure 2.10: Reverse translation[8]

The system finally updates the corresponding story card (fig. 2.11) in the target stack.

Figure 2.11: Translations into German

8 'The captain has brought back a Chinese vase.'

Chapter 3

Some aspects of LIDIA-1.0 in more detail

3.1. User interface

3.1.1. Preferences

There are four preference profiles, concerning the task, the communication and MT servers, the user, and the lexical resources (fig 3.1).

Task-related preferences are shown in the figure: the user selects the active target languages, the translation unit (selection, card or stack), and the desired component processes: spelling and style checking, terminological normalization, special fixed phrases, text categorization, and translation[9].

User preferences concern the feed-back type and the dialogue level. The lexical profile determines the active spellchecker, personal dictionaries, and thesaurii.

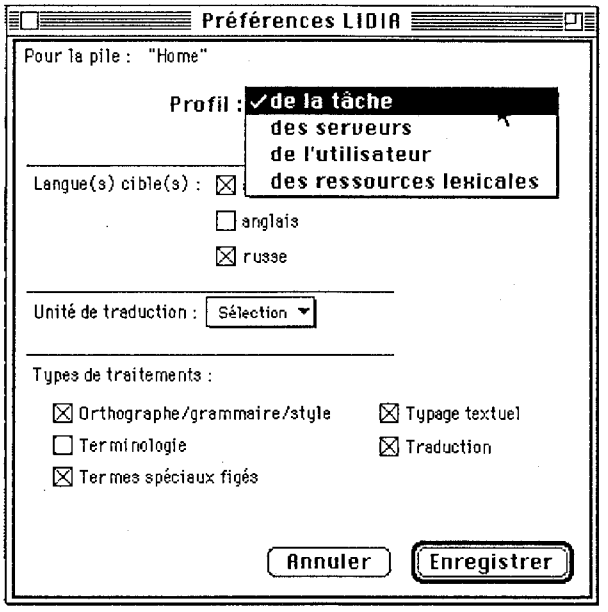


Figure 3.1: LIDIA-1 preferences

3.1.2. Menu

Once the preferences have been defined, the author uses a menu and a palette to interact with LIDIA.

The interaction with the author is made through the LIDIA menu (fig. 3.2), the Messages menu, a palette (fig. 3.3), feedback buttons (fig. 1.1) and windoids (fig. 1.1).

[9] The only component process now available in the mockup is translation.

The menu shown here offers 8 choices: process the selected object according to the set of preferences, process some object with a particular preference set, show the treatments' progress, show the reverse translation, show the annotations, show the palette, modify the preferences and build the target stacks.

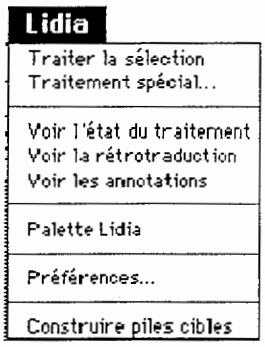


Figure 3.2: LIDIA-1 menu

3.1.3. Palette

The user can trigger the most frequent treatments by using the LIDIA-1 palette. The first line contains the LIDIA tools (process the selected object, show the treatment progress, show the annotations and show the reverse translation), and the second line the most frequent browsing tools.

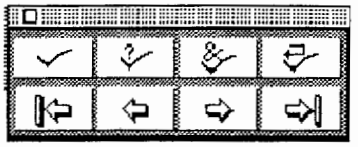


Figure 3.3: LIDIA-1 palette

3.2. Implementation issues

The implementation is characterized by the use of a distributed architecture, a whiteboard approach, and object-oriented techniques.

3.2.1. Distributed architecture

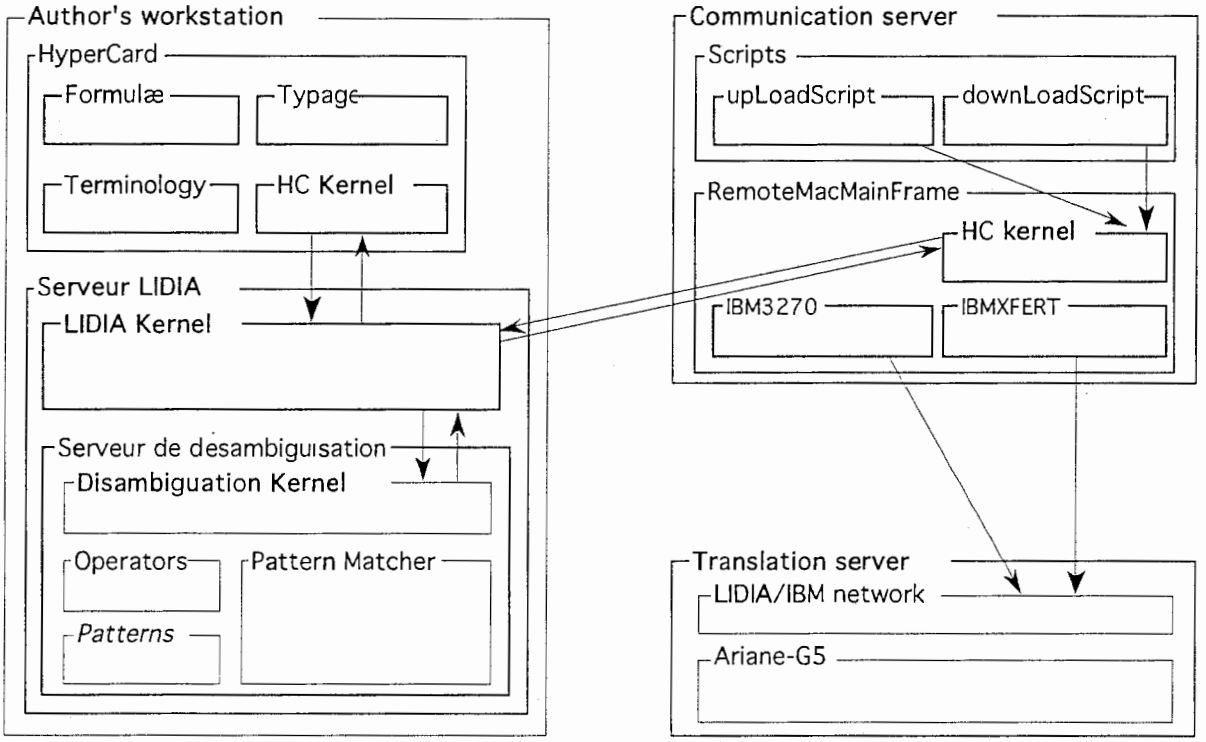


Figure 3.4: Architecture of LIDIA-1

Three machines (fig. 3.4) are involved in the translation process.

On the author's workstation the HyperCard Kernel sends and receives messages from the LIDIA kernel which organises the translation process for each object. The LIDIA Kernel sends translation jobs to the Translation server via the Communication server. The LIDIA Kernel also asks to prepare the disambiguation questions.

3.2.2. Whiteboard approach

For each object to be translated, the LIDIA Kernel creates a *mirror object* (a text file) which stores all information required by the translation process and necessary for the construction of the target stack. We distinguish between *static* and *dynamic information*. Static information is what is attached by HyperCard to each object. It is necessary to construct target stacks. Dynamic information is any information used by LIDIA to translate the content of an object.

These files can be considered to constitute a *whiteboard* as defined in [Boitet & Seligman 1994 ; Seligman & Boitet 1994]. Unlike the blackboard, the whiteboard is accessed only by a *coordinator* (the LIDIA Kernel), and not by the components (Disambiguation kernel and Ariane-G5). The main advantage of this architecture is to allow easy integration of existing or new components without having to modify them.

3.2.3. Object oriented techniques

All components but the lingware use object-oriented programming. The module for the Terminology, the idioms and the Typage as the kernel of the Communication server are written in HyperTalk, the HyperCard scripting language.

The LIDIA serveur is written in CLOS (MCL). Although encapsulated within the same environment, the LIDIA Kernel and the Disambiguation Kernel communicate by exchanging messages and might run on another machine.

Our use of messages and object-oriented programming techniques is close to the actor model used in the context of distributed cooperative systems.

Interactive disambiguation in some related work

Interactive MT was first proposed in the sixties by M. Kay for the MIND system [Kay 1973], and several projects experimented with variations of this design, notably the ITS project [Melby 1981 ; Melby 1987 ; Melby, *et al.* 1980] at Provo (BYU-TSI) from 1975 to 1981, the Alvey N-tran project [Chandler, *et al.* 1987 ; Whitelock, *et al.* 1986] at Manchester (UMIST-CCL) from 1985 to 1987, and the DLT project [Papagaaij 1986 ; Sadler 1989 ; Schubert 1986] at Utrecht (BSO) from 1982 to 1988. In KBMT-89 [Brown 1989 ; Brown & Nirenburg 1990 ; Carbonell 1983 ; Church & Patil 1982 ; Goodman & Nirenburg 1991 ; Tomita 1984 ; Tomita 1985 ; Tomita 1986] at CMU-CCL, questions were also asked by the "augmentor" if ambiguities could not be solved by the ontology. Ongoing work on 'MT for the target language inexpert' [Huang 1990] has been described at COLING-90. JETS, IBM-Japan's Japanese-English MT system, is based on an interactive Japanese dependency parser [Maruyama, *et al.* 1990 ; Tsutsumi, *et al.* 1993]. We can also mention the ITS project [Wehrli 1990 ; Wehrli 1991 ; Wehrli 1992 ; Wehrli 1993] at the university of Geneva and finally another project in Japan [Yamaguchi, *et al.* 1993].

We are of course indebted to many of these projects, which have explored several disambiguation contexts and strategies. However, if our information is correct, there are still no practical systems based on these projects. In some cases, we think potential users are confused by the fact that the disambiguator often produces questions in a somewhat unpredictable and esoteric way. This is why we try to build a system which has a simple structure, and asks questions understandable by college graduates.

In some other cases, we feel that the situation chosen is not appropriate for DBMT. For instance, JETS is currently tested on writers of technical documentation, who are not normally responsible for translation, and are accustomed to simply give their texts to professional translators. As a result, they are quite reluctant to use the system, although it is quite good and would certainly be very successful with employees of small firms having to produce similar material in Japanese and English and willing to help the machine translate on the spot rather than to pay and wait for professional translation.

For speech systems, the interactive clarification approach is also a solution as shown in [Frankish, *et al.* 1992] and proposed in [Ainsworth & Pratt 1992] and [Saito 1992].

Part II

Interactive disambiguation with LIDIA-1.0

Introduction

As far as the disambiguation is concerned, with the LIDIA-1 mockup, we solve the ambiguities calculated by the Ariane[10] analyzer used in the framework of the project. In this chapter we describe at first the multilevel, multisolution and concrete structure (mmc-structure) produced by the analyzer and then, the disambiguation process we propose [Blanchon 1994a]. Then, we will describe the meta-classes and the classes of ambiguity proper to the mmc-structure.

The disambiguation process is based on a pattern matching mechanism. Some of the ambiguity classes we have defined are associated with one or several sets of patterns. A set of patterns describes an occurring set of an ambiguity. A pattern describes a set of trees. The patterns of a set share several variables. An ambiguity **A** is recognized with the set of patterns **S** when:

- all the patterns of **S** match at least one of the multilevel and concrete structures,
- all the variables shared between the patterns of **S** have the same value.

The examination order of the sets of patterns is fixed by a strategy.

Some of the defined ambiguities are not solved by the pattern matching method. We think it is possible to solve them with the pattern matching mechanism but this solution has not been studied in great detail yet.

It has to be pointed out that the disambiguation process we propose is adaptable to any analyzer producing multiple and concrete tree structures.

[10] A full and concise description of Ariane is given in [Boitet 1990a]

Chapter 4

Meta-classes & classes of ambiguity

A linguistic classification of the ambiguities is given in Appendix A, as a reference. All the ambiguity classified and described in this chapter have to be considered in the disambiguation framework. This classification has been established according to pragmatical reasons.

We define four *meta-classes* of ambiguity in the framework of the disambiguation process:

- lexical ambiguity meta-class
- categorial ambiguity meta-class
- geometrical ambiguity meta-class
- labeling ambiguity meta-class

These meta-classes are related to the linguistic ambiguities in this way:

- the lexical ambiguity meta-class covers the ambiguities defined in §A.2.1.,
- the categorial ambiguity meta-class covers the ambiguities defined in §A.2.2.,
- the geometrical ambiguity meta-class covers the ambiguities defined in §A.3.2.,
- the labeling ambiguity meta-class covers the ambiguities defined in §A.4.

Let's see how the chosen meta-classes are refined into classes of ambiguity.

4.1. Lexical ambiguity

There is a *lexical ambiguity* if the analyzer has not been able to choose one and only one word sense for a word occurring the the analyzed sentence.

4.2. Categorial ambiguity

There is a *categorial ambiguity* if the analyzer can not attach one and only one syntactic class to each occurrence (word) of the sentence. Then it results that the vectors obtained after the projection of the syntactic classes of the occurrences are different.

We distinguish two classes of categorial ambiguity:

4.2.1. Verbal coordination

If a coordinated phrase can be interpreted as a noun phrase or a verbal phrase then we will say that this phrase introduces an ambiguity of *verbal coordination* in the sentence.

The next figure (figure 4.1, labels a given in Appendix C) shows the two trees produced for the sentence "`Il observe la photo et la classe`[11]."

In the first analysis (upper tree), 'classe' is the verb 'classer', its subject is the subject of the main phrase ('il') and 'la' is a pronoun replacing the object of the main phrase ('la photo') — Il observe la photo et il classe la photo[12].

In the second analysis (lower tree), "et la classe" is a noun phrase coordinated to the verb "observer" — Il observe la photo et il observe la classe[13].

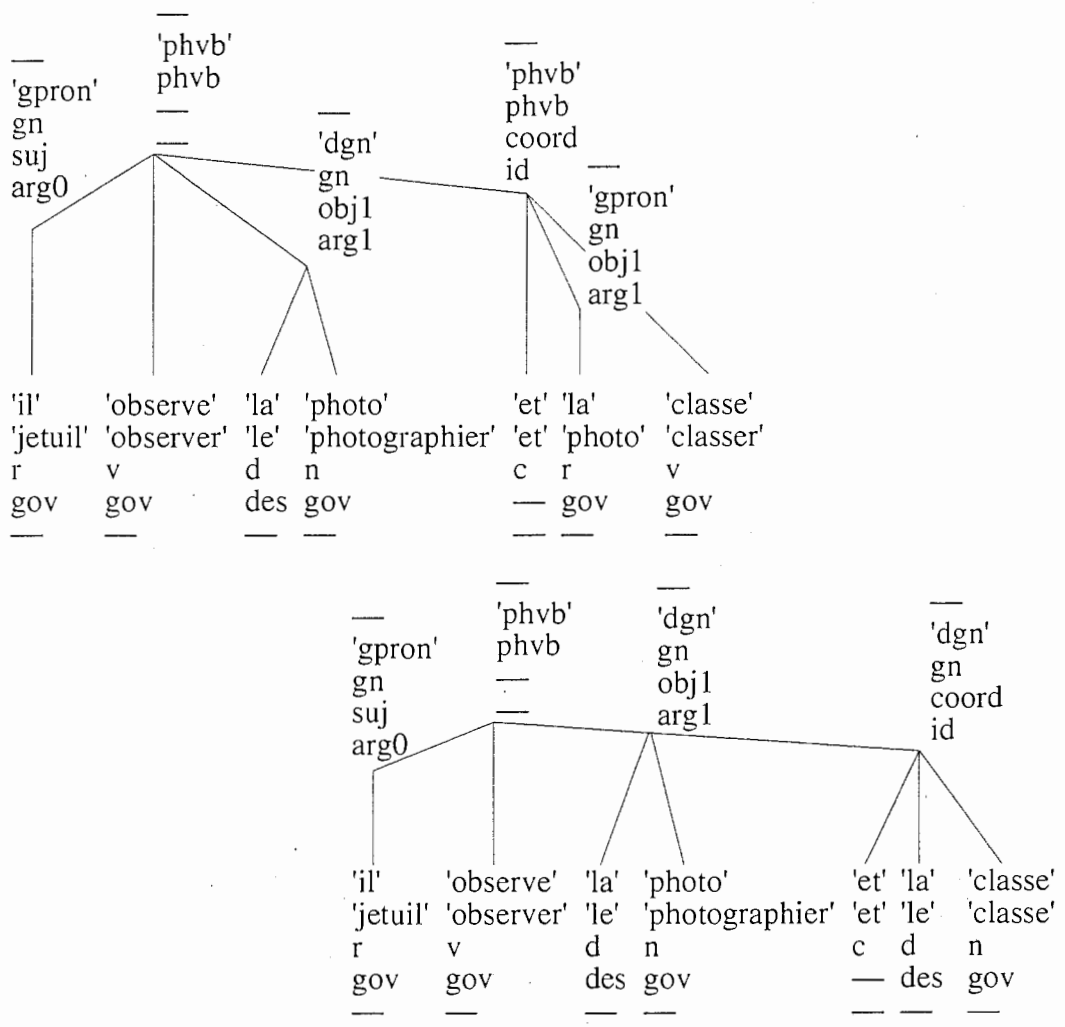


Figure 4.1: Verbal coordination ambiguity

4.2.2. Non-verbal coordination

All the categorial ambiguities that are not verbal coordination ambiguity are said to be non-verbal coordination ambiguities.

---

[11] He observes the photo and the (it) classroom (files).

[12] He observes the photo and files it.

[13] He observes the photo and he observes the classroom.

The next figure (figure 4.2) shows the two trees produced for the sentence "`Devant cette somme, il hésite`[14]."

"Devant" can be interpreted as the gerund form of the verb 'devoir' (upper tree) — Il hésite parce qu'il doit cette somme[15]. — or as a preposition (lower tree) — En face de cette somme, il hésite[16].

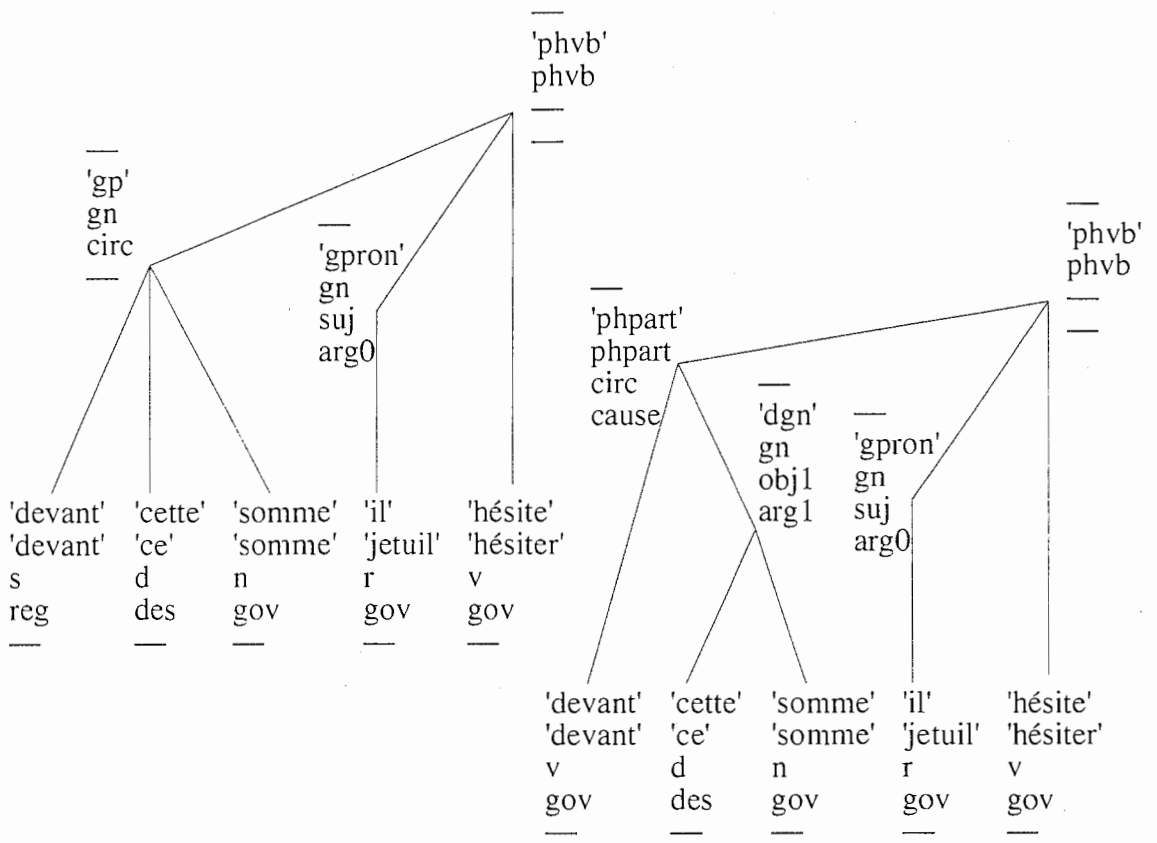


Figure 4.2: Non-verbal coordination ambiguity

4.3. Geometrical ambiguity

There is a *geometrical ambiguity* if the structures produced in the multiple structure are different and the cause of this difference is not a categorial ambiguity. This means that a single syntactic class has been determined for each occurrence.

Sub-categories of the geometrical ambiguity are defined like this:

4.3.1. Argument structure of the verb

If the verb of a sentence can have several constructions we will speak about an ambiguity of *argument structure of the verb* (we can also have that kind of problem with the trees having the same geometry, cf. infra).

The next figure (figure 4.3) shows the three trees produced for the sentence "`Il parle de l'école de cuisine lyonnaise`[17]." The verb 'parler' can be complemented:

---

[14] In front of (owing) this amount, he hesitates.
[15] He hesitates because he ows this amount.
[16] In front of this amount, he hesitates.
[17] He speakes from (about) the school Ø or about (Ø)cooking. (word by word translation)

- with a first argument (upper tree) — Il parle à propos de l'école de cuisine[18],
- with a circumstant (lower tree) — Il parle depuis l'école de cuisine[19],
- with a first argument and a circumstant (middle tree) — Il parle depuis l'école à propos de cuisine[20].

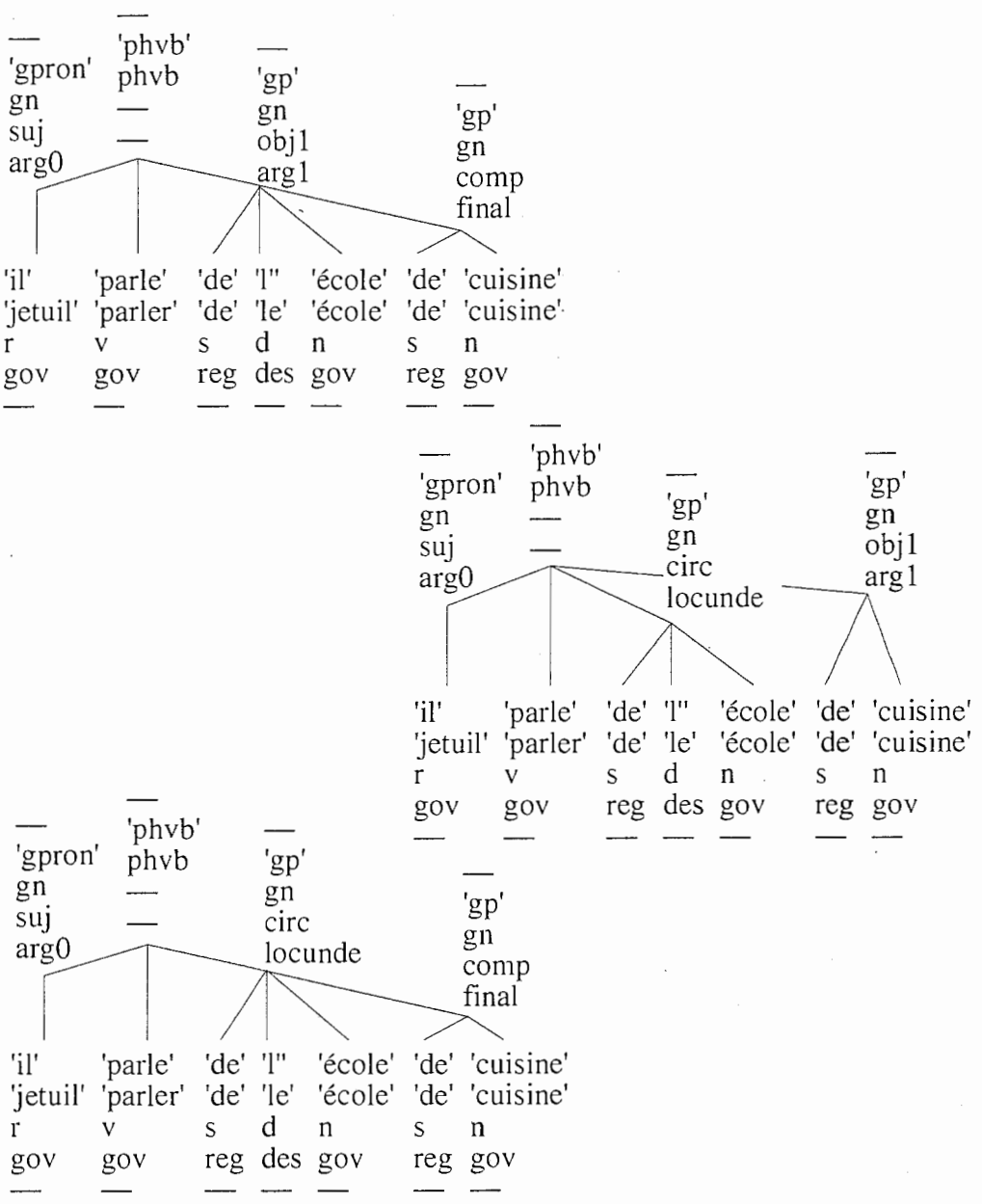


Figure 4.3: Ambiguity of argument structure of the verb

4.3.2. Coordination

The ambiguity of *coordination* is subcategorized into two types of ambiguity:

[18] He speaks about the cooking school.
[19] He speaks from the cooking school.
[20] He speaks from the school about cooking.

i. *Noun coordination*

In this case a coordinated noun phrase can be attached to several precedent phrases in the sentence.

The next figure (figure 4.4) shows the two trees produced for the sentence "`L'évolution de la structure du réseau et des investissements est étudiée`[21]."

On one hand, the noun phrase "et des investissements" can be attached to "de la structure" — l'évolution de la structure des investissements est étudiée[22] (upper tree).

On the other hand, the noun phrase "et des investissements" can also be attached to "l'évolution" — l'évolution des investissements est étudiée[23] (lower tree).

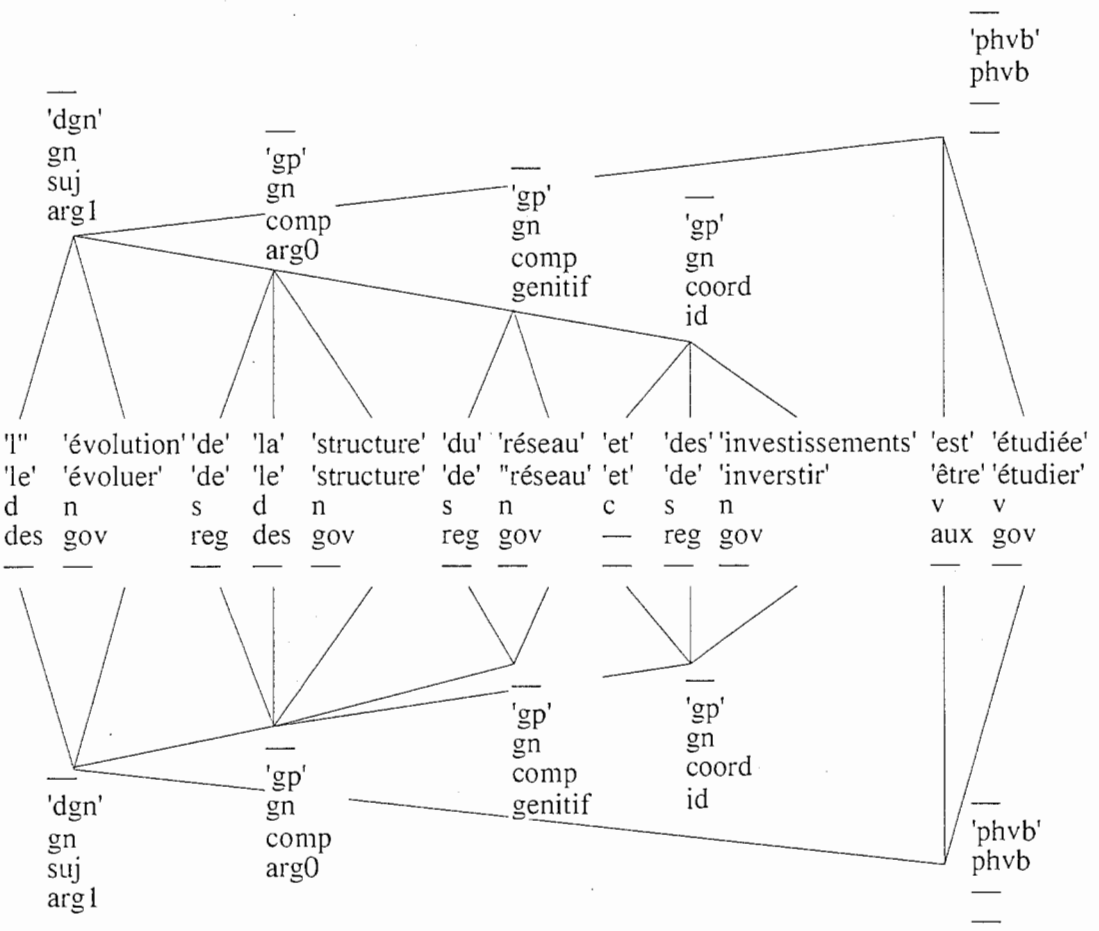


Figure 4.4: Noun coordination ambiguity

ii. *Adjective coordination*

It can also be necessary to know the distribution of an adjective phrase over a set of coordinated noun phrases.

The next figure (figure 4.5) shows the two trees produced for the sentence "`Il prend les cahiers et les classeurs noirs`[24]."

---

[21] The evolution of the structure of the network and the investments is studied.
[22] The evolution of the structure of the investments is studied.
[23] The evolution of the investments is studied.
[24] He takes notebooks and files black. (word by word translation)

On one hand (upper tree), "noirs" has to be distributed only on the noun phrase "et les classeurs" — Il prend les classeurs noirs et les cahiers[25].

On the other hand (lower tree), "noirs" has to be distributed on the noun phrases "les cahiers" and "et les classeurs" — Il prend les cahiers noirs et les classeurs noirs[26].

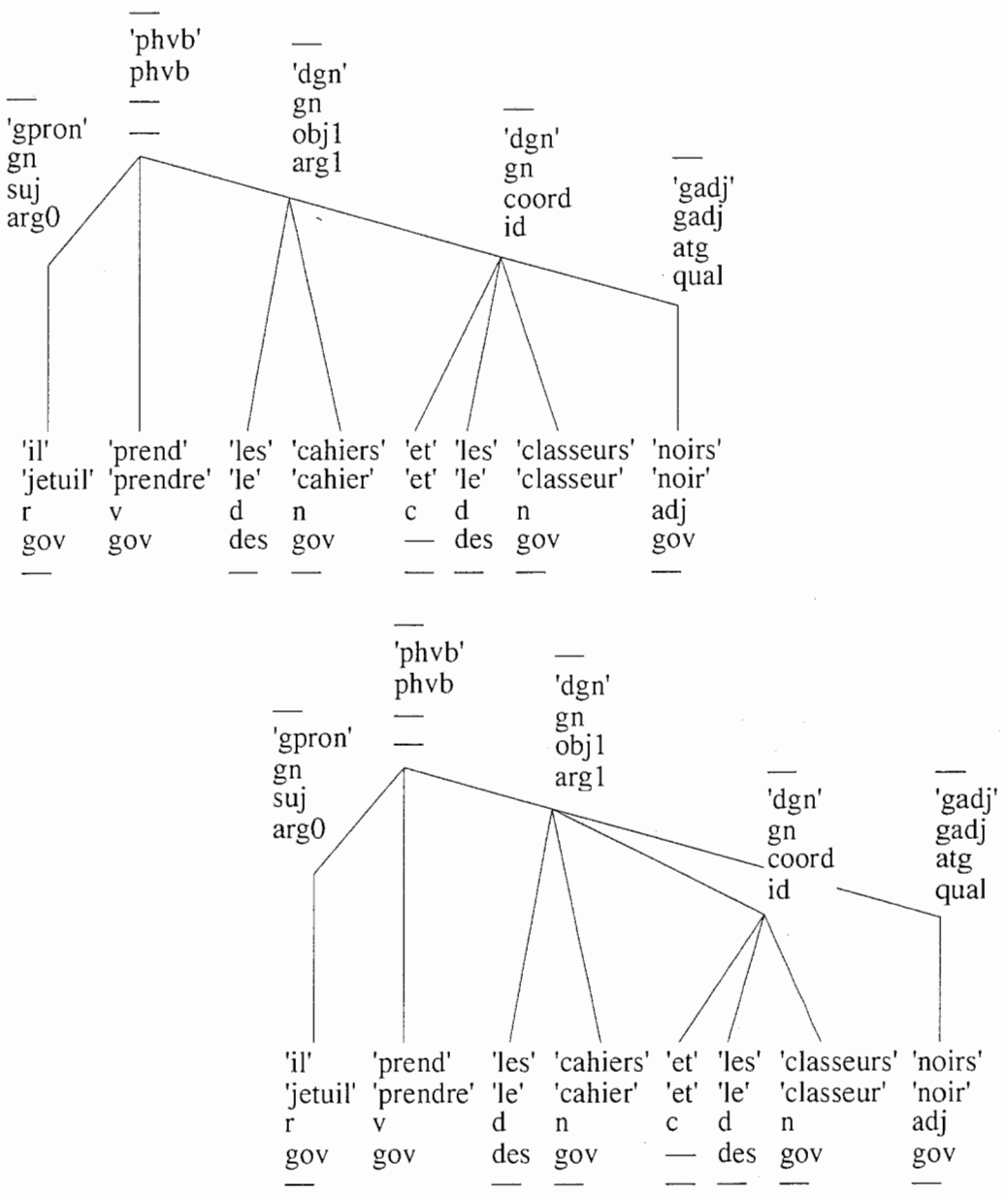


Figure 4.5: Adjective coordination ambiguity

4.3.3. Subordination

When a phrase can attached to several previous phrases (not coordinated), we have an ambiguity of *subordination*.

The next figure (figure 4.6) shows the two trees produced for the sentence "`Elle épouse un professeur de droit anglais`[27]."

---

[25] He takes (black files) and notebooks.
[26] He takes (black notebooks) and (black files).
[27] She maries a teacher of law english. (word by word translation)

On one hand (upper tree), "anglais" has to be attached to the noun phrase "de droit" — Elle épouse un professeur qui est spécialiste du droit anglais[28].

On the other (lower tree), "anglais" has to be attached to the noun phrase "un professeur" — Elle épouse un professeur anglais spécialiste de droit[29].

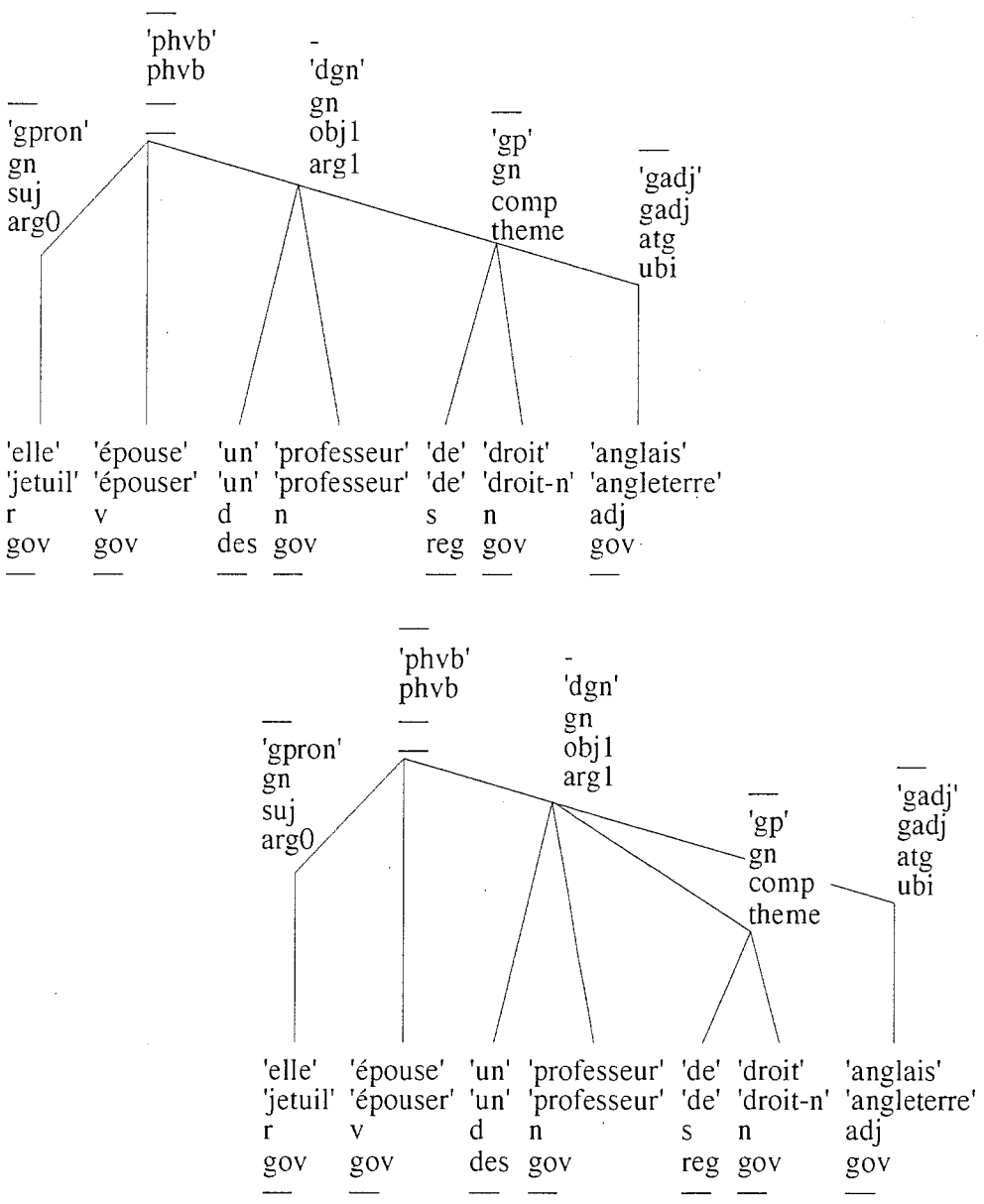


Figure 4.6: Subordination ambiguity

4.4. Labeling ambiguity

There is *labeling* ambiguity if, when the geometry of the structure is determined, the analyzer can not choose a unique set of labels for all the nodes of the tree.

[28] She maries an English law teacher.
[29] She maries an English teacher of law.

4.4.1. Logico-semantic labeling

When the analyzer can not calculate only one labeling in logico-semantic relations for the tree, we will speak about an ambiguity of *logico-semantic labeling*.

The next figure (figure 4.7) shows the two trees produced for the sentence "`Pierre fait porter des chocolats à Lucie`[30]."

On one hand (upper tree), the noun phrase "à Lucie" is the third argument (marked arg2) of the verb "porter" — Lucie est la destinataire de chocolats envoyés par Pierre[31].

On the other (lower tree), the noun phrase "à Lucie" is the first argument (marked arg0) of the verb "porter" — Suivant les consignes de Pierre, Lucie porte des chocolats[32].

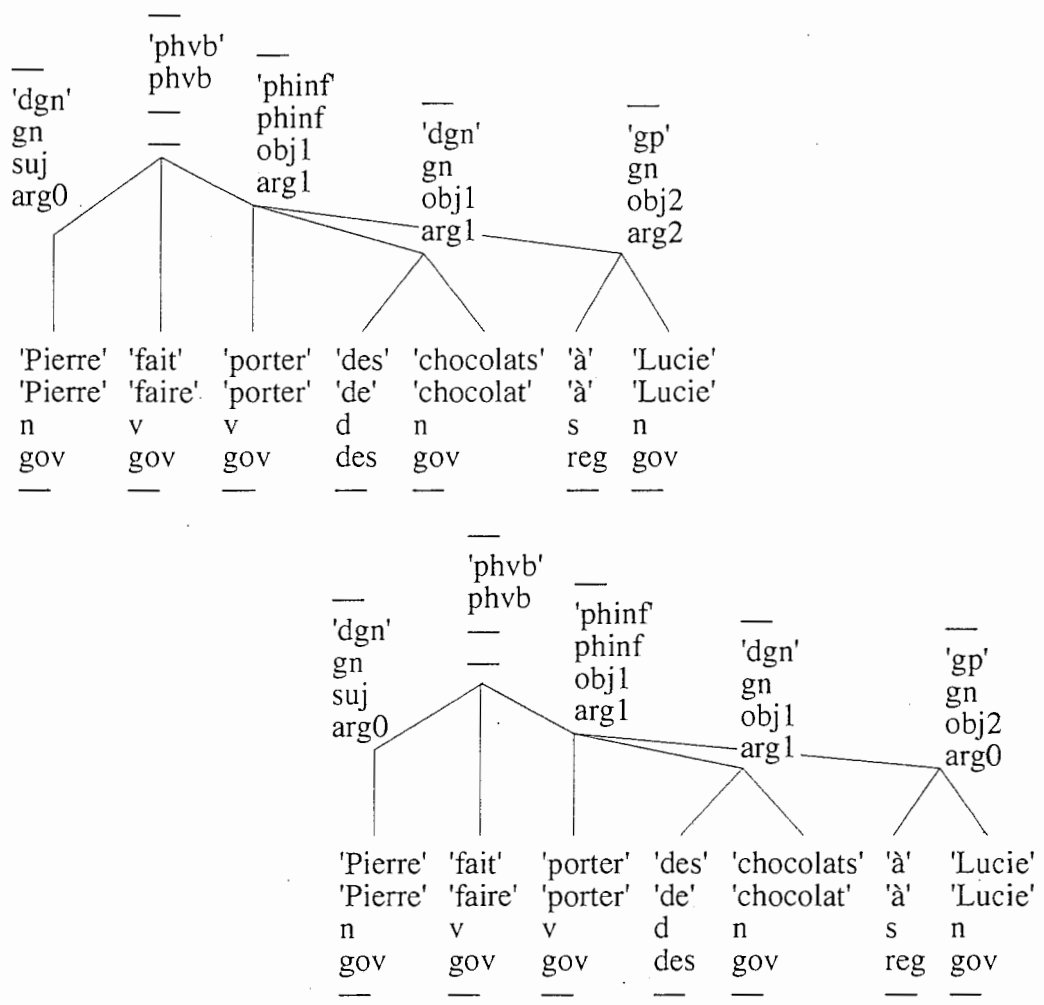


Figure 4.7: Logico-semantic labeling ambiguity

4.4.2. Syntactic labelling

When the analyzer can not calculate only one labelling in syntactic function for the tree, we will speak about an ambiguity of *syntactic labelling*.

---

[30] Pierre is having sent chocolates to (by) Lucie. (word by word translation)
[31] Pierre is having chocolates sent to Lucie.
[32] Pierre is having chocolates sent by Lucie.

The next figure (figure 4.8) shows the two trees produced for the sentence "`Je vous parle de la Tour Eiffel`[33]."

On one hand (upper tree), the noun phrase "de la Tour Eiffel" is the first object of the verb "parler" — Je vous parle à propos de la Tour Eiffel[34].

On the other hand (lower tree), the noun phrase "de la Tour Eiffel" is a circumstant of the verb "parler" — Je vous parle depuis la Tour Eiffel[35].

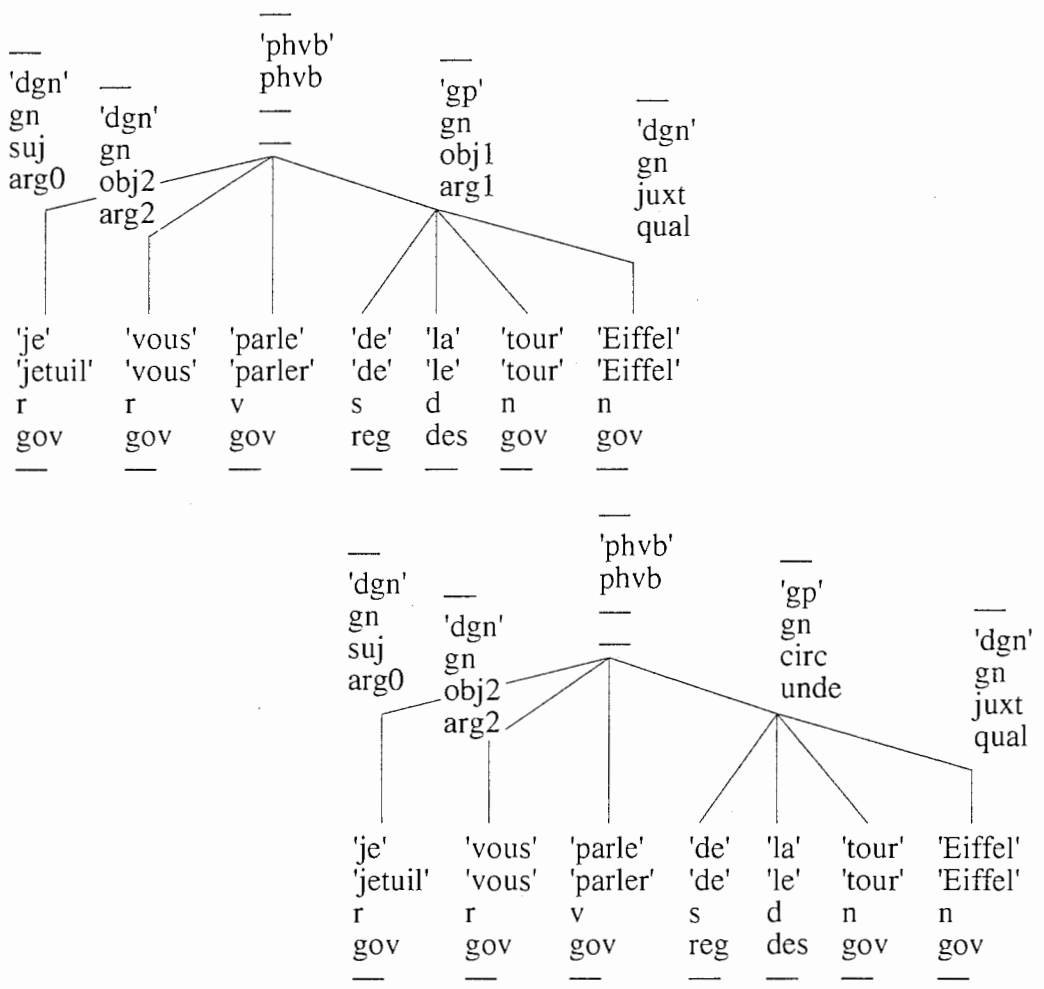


Figure 4.8: Syntactic ambiguity labeling

4.4.3. Argument order for a direct transitive verb

When the analyzer did not calculate the order of the arguments for a direct transitive verb we will speak about an ambiguity of *argument order for a direct transitive verb*.

The next figure (figure 4.9) shows the two trees produced for the sentence "`Quel auteur cite ce conférencier ?`[36]".

---

[33] I you speak from (about) the Tower Eiffel. (word by word translation)
[34] I speak to you about the Eiffel Tower.
[35] I speak to you from the Eiffel Tower.
[36] Which author quotes this lecturer? (word by word translation)

On one hand (upper tree), the noun phrase "quel auteur" is the subject of the verb "citer" and the noun phrase "ce conférencier" is the first object of the verb — Un auteur parle de ce conférencier[37].

On the other (lower tree), the noun phrase "quel auteur" is the first object of the verb "citer" and the noun phrase "ce conférencier" is the subject of the verb — Ce conférencier parle d'un auteur[38].

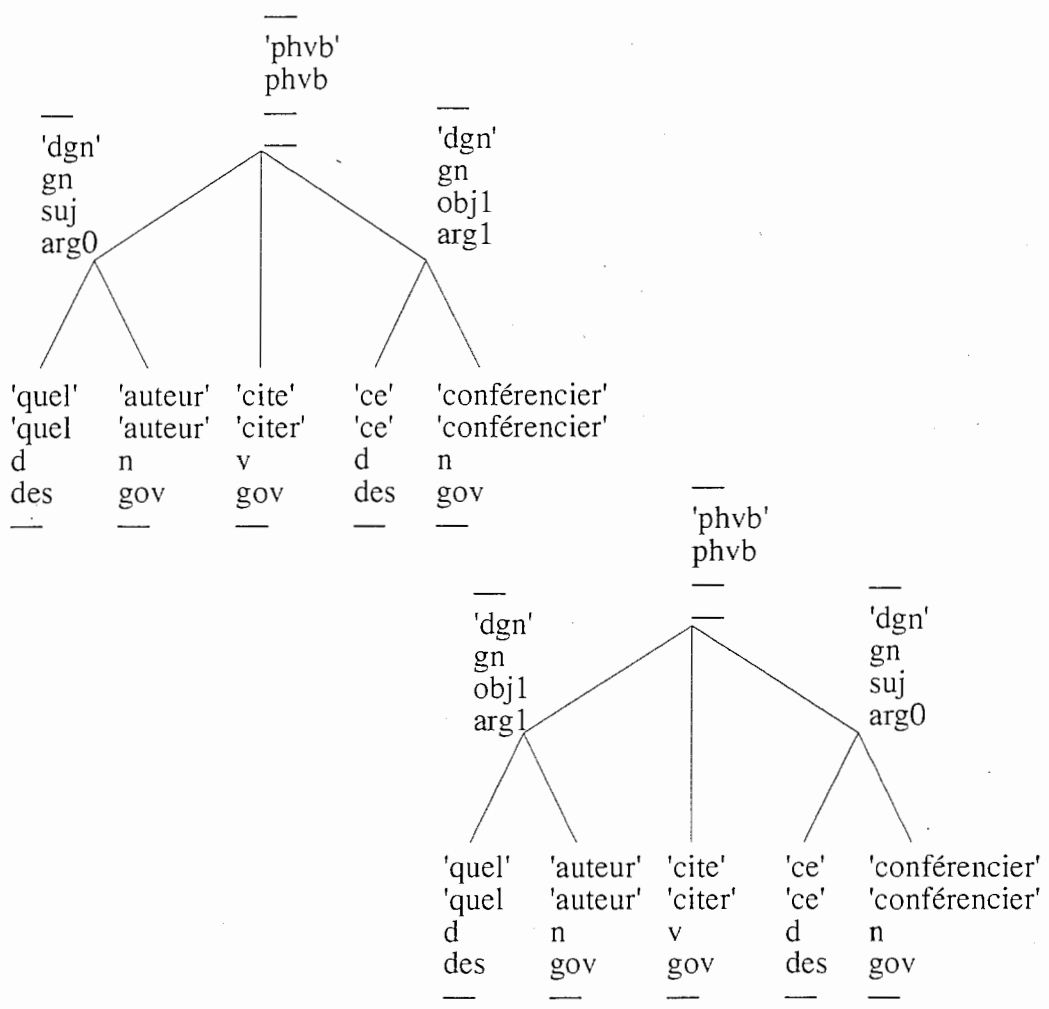


Figure 4.9: Ambiguity of argument order of a birect transitive verb

[37] An author is quoted by this lecturer.
[38] This lecturer is quoted by an author.

Chapter 5

Main characteristics of the disambiguation process

The order in which the ambiguities are solved, if more than one appear in a sentence, is defined by a strategy. This strategy is imposed according to four principles. A method to produce a dialogue item is associated with each pattern. Thus, once an ambiguity is recognized the system can produce a clarification dialogue. The dialogue item construction methods are described by means of a set of basic operators.

5.1. Principles

The four principles we used to define the strategy are listed below in the order in which they must be satisfied:

1 first, find the right simple phrases,

2 second, find the construction of the verbs,

3 third, find the structure of the dependents of the verbs,

4 last, find the word senses.

Pragmatic considerations led us to define them:

- The simple phrases are the basic bricks of the sentence, producing the sense;
- Then, we want to find the constituents of the sentence at the higher level, the level of the construction of the verb;
- After that, we want to find the right organization for the constituants of the verbs;
- Finally, it is time to find the sense of each occurrence as the whole set of senses of an occurrence has been reduced once the previous steps allowed to find its use in the sentence.

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

5.2. Strategy

To be consistent with the principles we have implemented the strategy described in the following flow chart.

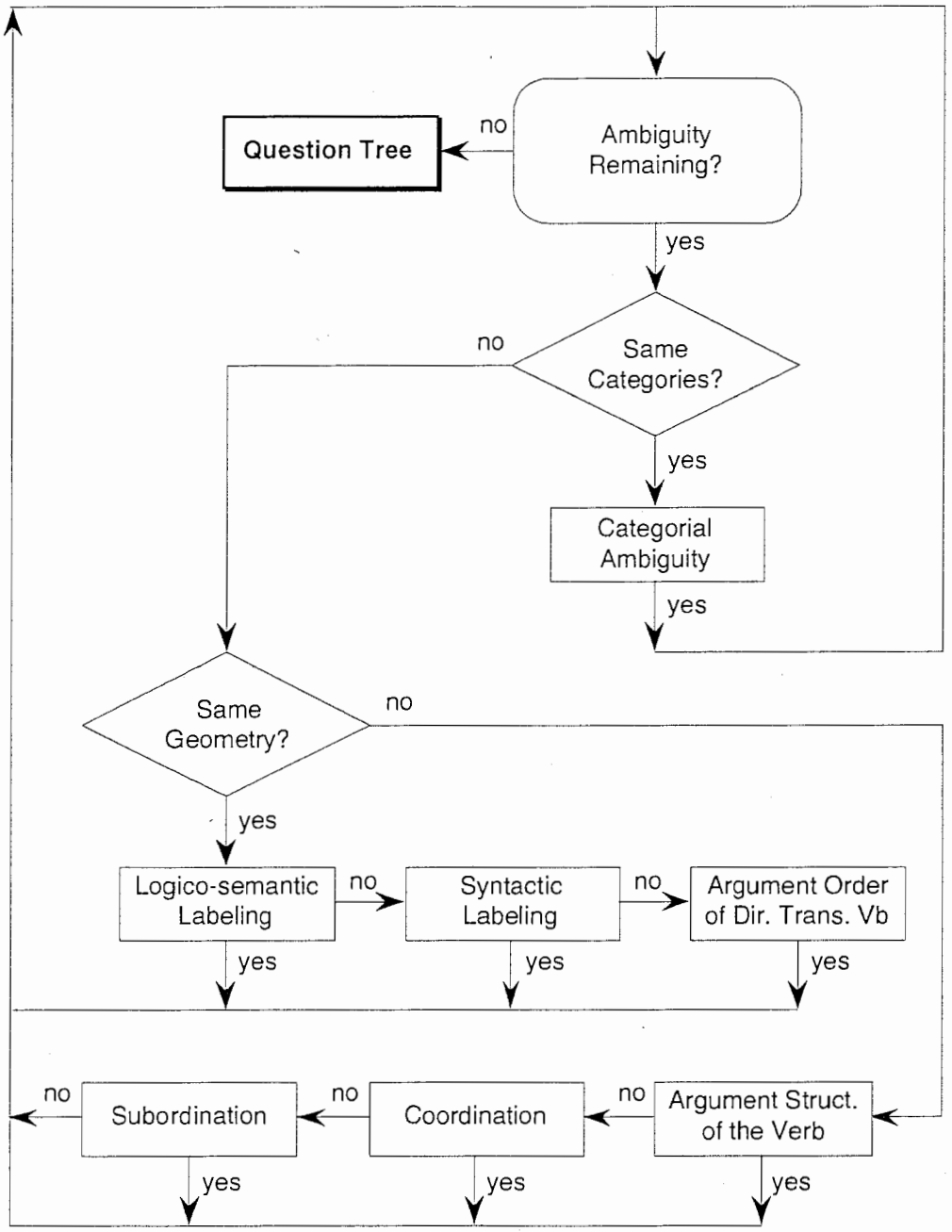


Figure 5.1: Disambiguation tree production strategy

A more concrete view of the creation of the question tree is given beneath (fig. 5.2):

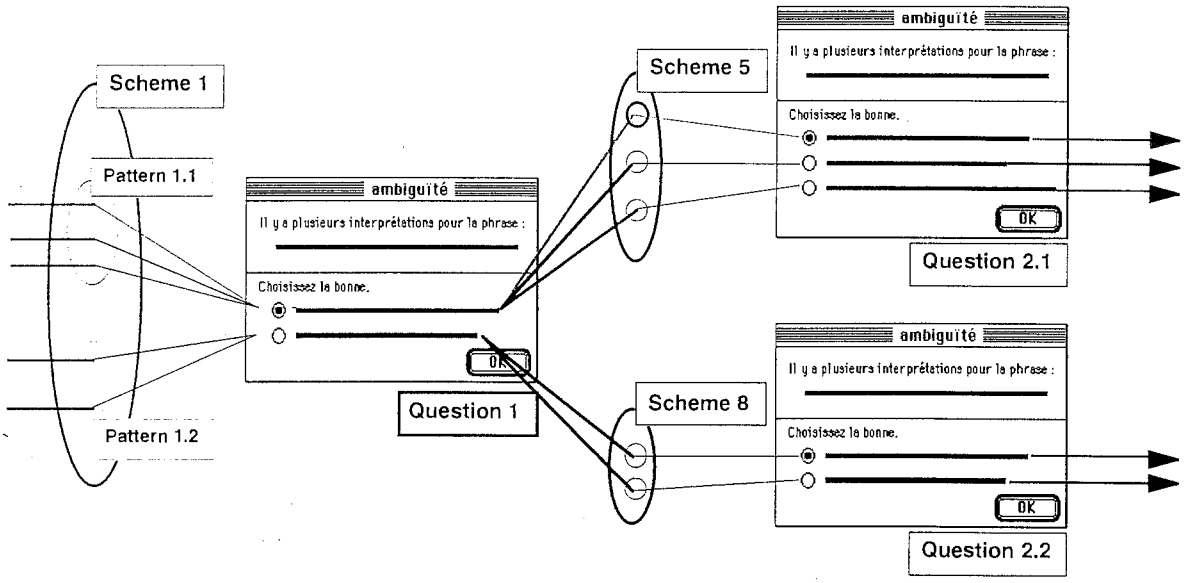


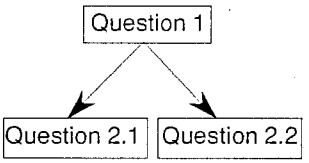
Figure 5.2: The construction of a disambiguation tree

In the previous chart we can see how the disambiguation process produce one question tree. In the example, the input is a five solution analysis.

The Scheme 1 of patterns is matched against the 5 solutions; the 3 first ones against pattern 1.1 and the 2 others against pattern 1.2. Thus the system produces a two items dialogue allowing to choose the right group (**Question 1**).

The process is called again with the two sets of analysis as the input. The first set is matched against Scheme 5 and the second set against Scheme 8. Two dialogues are then produced: one three item dialogue for the first set (**Question 2.1**) and one two item dialogue for the second (**Question 2.2**).

The question tree is made of the 3 questions :



5.3. Techniques, operators and methods

The disambiguation dialogue items are produced using simple techniques involving the use of a set of basic operators. The dialogue item production method associated with each patterns of a set allow us to build one disambiguation dialogue per set.

5.3.1. Techniques

We choose not to use an automatic generator to produce the dialogue items because we need to be able to ensure that we are not producing ambiguous dialogue. This kind of tool must not be used for such a task.

Indeed, we should ensure that the dialogue items are not ambiguous and that they are all different, but also different from the original sentence.

The techniques we have chosen are simple but we hope effective:

- the display of dictionary definitions or synonyms,
- the display of the syntactic classes of the occurrences,

- the reformulating of elided phrases,
- the permutation and bracketing of phrases,
- the semantic replacement of ambiguous prepositions.

5.3.2. Operators

To produce the dialogue items, we have defined the following operators that operate on parts of the mmc-structure. Those parts of the mmc-structures are next called linguistic trees. It can be a tree or a forest of trees.

`Agreement` produce the form of an adjective according to gender and number constraints.

```
ex :    Agreement("noirs",(("gn" [...] (gnr fem nbr plu)) ([...]))) -> "noires"
```

`Text` produce the text of the linguistic trees given as parameter.

`Subject` produce the text of the subject of the linguistic trees given as parameter.

– **note:** There is such a function for each syntactic function and syntagmatic class. Example: `VerbalGroup`, `CircComp`...

`Coord` produce the coordinating occurrence of the linguistic trees given as parameter.

`But_Coord` produce the text of the linguistic trees given as parameter without the coordinating occurrence.

`But_Sub` produce the text of the linguistic trees given as parameter without the subordinating particle.

`But_Det` produce the text of the linguistic trees given as parameter without the determiner.

`Distribute` distribute an occurrence or a groupe of occurrences over other groups of occurrences and link the new groups with a preposition of coordination.

– **note:** The distribution is done agreeing the gender and the number of the adjective with the gender an the number of the different substentives

```
ex :    Distribute((A, B C, D), ou ,(1, 2), (1, 3)) -> A B C ou A D
```

`Substitute` replace an ambiguous preposition by a non-ambiguous one (in the context) according to several properties: syntactic function or logico-semantic relation.

```
ex :    Substitute("de", #Objet 1) -> "à propos de"
```

`Project` applied to a leaf of the mmc-structure, project the occurrence and the syntactic class of the occurrence. In the future some other information may also be proposed.

`Bracket` brackets the text of the arguments.

```
ex :    Bracket("classeurs" ,"noirs") -> "(classeurs noirs)"
```

`Definition` produce the definition of the occurrence given as the parameter.

5.3.3. Methods, informal view

In this paragraph we will give a non formal view of the method associated with each defined classe of ambiguity and used to solve the problem.

i. Lexical ambiguity

A lexical ambiguity is solved by presenting to the user the word senses selected by the analyzer. (while experimenting we found that this solution was not a good one; synonyms or examples should give better usability).

ii. Categorial ambiguity

Verbal coordination

A verbal coordination ambiguity is solved reformulating the possible elisions reordering some phrases of the original sentence. Two kinds of elision are possible:

- The coordinated phrase is a noun group. This phrase is the object of the last encountered verb in the sentence, the subject of the verb does not change.

  For the sentence "Il atteint la grange et la ferme[39]," we get "Il atteint la ferme[40]."

- The coordinated phrase is made of a pronoun and a verb. The subject and the object of the verb have been elided. The object (the pronoun) is the object of the last encountered verb in the sentence. The subject is the subject of the last encountered verb.

  For the sentence "Il attein la grande et le ferme," we get "Il ferme la grange[41]."

Non-verbal coordination

A non-verbal coordination ambiguity is solved showing to the user the syntactic class calculated by the analyzer of each ambiguous occurrence.

For the sentence "Le pilote ferme la porte[42]," we get "le pilote(nom) ferme(adjectif) la(pronom) porte(verbe)[43]." and "le pilote(nom) ferme(verbe) la(article) porte(nom)[44]."

iii. Geometrical ambiguity

Argument structure of the verb

An ambiguity of argument structure of the verb is solved replacing the ambiguous prepositions by non ambiguous ones in the context.

For the sentence "Il parle de l'école de cuisine[45]," we get "Il parle depuis l'école de cuisine[46].", "Il parle depuis l'école à propos de cuisine[47]." and "Il parle à propos de l'école de cuisine[48]."

Coordination

An ambiguity of coordination is solved distributing the phrase which the coordination left out. We are thus able to show the user all the possible distributions.

With noun coordination, a noun phrase can have several attachments with its preceding phrases in the sentence.

For the sentence "On étudie l'évolution de la structure du réseau et des investissements[49]," we get "On étudie l'évolution de la structure des investissements[50]" or "On étudie l'évolution des investissements[51]" or "On étudie des investissements[52]."

---

39 He reaches the barn and the (he) farm (closes). (word by word translation)
40 He reaches the farm.
41 He closes the barn.
42 The pilot shuts (firm) the (her) door (carries). (word by word translation)
43 The firm pilot carries her.
44 The pilot shuts the door.
45 He speaks from (about) the school Ø or about (Ø) cooking. (word by word translation)
46 He speaks from the cooking school.
47 He speaks from the school about cooking.
48 He speaks about the cooking school.
49 We study the evolution of the structure of the network and the investments. (word by word translation)
50 We study the evolution of the structure of the investments.
51 We study the evolution of the investments.
52 We study the investments.

With adjective coordination, an adjective can be distributed on different coordinated noun phrases.

Subordination

An ambiguity of subordination is solved permuting the subordinated phrases which have been separated by an intermediate one. We use brackets to show the phrases we have grouped together.

For the sentence "L'école de cuisine lyonnaise ouvre ses portes[53].", we get "l'école de (cuisine lyonnaise)[54]" or "l'(école lyonnaise) de cuisine[55]."

iv. Labeling ambiguity

Logico-sémantique

An ambiguity of logico-sémantique labeling is solved by replacing the ambiguous prepositions by non ambiguous ones in the context of the sentence. Those new prepositions present all the possible labelling.

For the sentence "Pierre fait porter des chocolats à Lucie[56].", we get "Pierre fait porter des chocolats pour Lucie[57]" or "Pierre fait porter des chocolats par Lucie[58]."

Syntactic labelling

An ambiguity of syntactic labelling is solved by replacing the ambiguous prepositions by non ambiguous ones in the context of the sentence. Those new prepositions present all the possible labelling.

For the sentence "Je parle de la Tour Eiffel[59].", we get "Je parle depuis la Tour Eiffel[60]." or "Je parle à propos de la Tour Eiffel[61]."

Argument order for a direct transitive verb

An ambiguity of argument order for a direct transitive verb is solved by showing the interpretations of the sentence in a standard order. This standard order is subject verbe object. The subject and object are constructed with the head (governor) of each ambiguous phrase associated with the right article.

For the sentence "quel hôtel gère ce bureau ?[62]", we get "un hôtel gère un bureau[63]" or "un bureau gère un hôtel[64]."

---

[53] The school of cooking lyonnaise open its doors. (word by word translation)
[54] The school of lyonnaise cooking.
[55] The lyonnaise school of cooking.
[56] cf. note 22
[57] cf. note 23
[58] cf. note 24
[59] I speak from (about) the Tower Eiffel. (word by word translation)
[60] I speak from the Eiffel Tower.
[61] I speak abourt the Eiffel Tower
[62] Which hotel manage this office? (word by word tranlation)
[63] A hotel manages an office.
[64] An office manages a hotel.

Chapter 6

Realization & results

In this last chapter we are going to explain how the disambiguation dialogues are produced. Examples, illustrating all the situations we have defined, conclude the document.

6.1. Patterns and pattern matching

The disambiguation process has to be seen as a filtering process. Most of the ambiguities are described by means of one or several sets of patterns. The patterns describe a set of trees. Each one is synchronized with the other by means of variables.

6.1.1. General framework

In the current implementation, geometrical ambiguities and verbal coordination ambiguities are described by several sets of patterns. Those patterns are described with a pattern description language we have defined. The ambiguities of decoration and non-verbal coordination are recognized by a filtering process applied on lists of node decoration.

An ambiguity **A** is recognized with the set of patterns **S** when:

- each pattern of **S** matches at least one multilevel and concrete structure,
- all the variables shared by the patterns of **S** have the same value.

Of course, it should be pointed out that all the kinds of ambiguities mentioned here are candidate to be solved by the pattern matching mechanism.

6.1.2. Patterns description language

Here is a BNF description of the pattern description language.

i. **Grammar**

```
pattern        ::=  variable              |
                    ; match every expression
                    constant              |
                    ; match only the atom (constant)
```

		segment-pattern |
		; match a segment
		simple-pattern |
		; match an expression
		(pattern . pattern)
		; recognize the first characterthen the rest
simple-pattern	::=	(?is variable pred args) |
		; check the predicate pred (cf ii. comments on the grammar)
		(?or pattern...) |
		; match one of the different character
		(?and pattern...) |
		; match all the patterns
		(?not pattern...)
		; match if the patterns are not recognized
segment-pattern	::=	((?* variable) ...) |
		; match zero or more expressions
		((?+ variable) ...) |
		; match one or more expression
		((?? variable) ...) |
		; match zero or one expression
		((?if expression) ...)
		; check if the expression (which may contain variables) is true
variable	::=	?character^{+}
		; ? followed by letters
constante	::=	atom
		; an atom

ii. Comments on the grammar

For the patron-simple "(?is variable pred args)" the predicate "pred" is applied on the arguments "variable" and "args" namely: pred (variable args).

iii. Examples

```
;;;-----------------------------------------------------------------------
;;;
;;; Title       : Pattern-matcher
;;;
;;;-----------------------------------------------------------------------
;;; Author      : Mathieu Lafourcade
;;; Copyright   : 1994 GETA, All Rights Reserved
;;; Availability: SAMBRE & LIDIA projects
;;;-----------------------------------------------------------------------
;;; Filename : pat-match.lisp
;;; Version  : 1.0
;;; Examples : (pat-match '(a (?* ?x) d) '(a b c d))
;;;          :  -> ((?X B C))
;;;          : (pat-match '(a (?* ?x) (?* ?y) d) '(a b c d))
;;;          : -> ((?Y B C) (?X))
;;;          : (pat-match '(a (?* ?x) (?* ?y) ?x ?y) '(a b c d (b c) (d)))
;;;          : -> ((?Y D) (?X B C))
;;;          : (pat-match '(?x ?op ?y is ?z (?if (eql (?op ?x ?y) ?z))) '(3 + 4 is 7))
;;;          : -> ((?Z . 7) (?Y . 4) (?OP . +) (?X . 3))
;;;          : (pat-match '(?x ?op ?y (?if (?op ?x ?y))) '(4 > 3))
;;;          : -> ((?Y . 3) (?OP . >) (?X . 4))
```

```
;;;          : (setf (get 'lidia-patterns 'cv-1)
;;;          :       '((?is ?x node-prop-equal-p 'CS 'PHVB)
;;;          :         (?* ?w)
;;;          :         ((?is ?y node-prop-equal-p 'FS 'OBJ) (?* ?p1))
;;;          :         ((?is ?z node-prop-equal-p 'CS 'PHVB) (?* ?p2))
;;;          :         ?punc
;;;          :         )
;;;          :       )
;;;          :
;;;          : (match (get 'lidia-patterns 'cv-1)
;;;          :       (get-solution (get 'lidia-test 'boule) 2)
;;;          :       )
;;;          : -->
;;;          : (
;;;          : (?PUNC ("." "." (CAT P)))
;;;          : (?P2 (("et" "ET" (FS REG CAT C))) ((NIL "GPRON" (K GN FS OBJ RL ARG1
;;;          :  CAT R GNR FEM PERS 3 NBR SING VALET N)) (("la" "BOULE" (FS GOV CAT
;;;          :  (R N) GNR FEM PERS 3 NBR SING SENS 1)))) ((NIL "NV" (K PHVB FS COORD
;;;          :  RL ID CAT V ENONCP DECL VOIX ACT MT IPR PHASE NONACC SUBV VF PERS 3
;;;          :  LINKS OUI NBR SING VALET Q ARGS A1)) (("lance" "LANCER" (FS GOV CAT V
;;;          :  MT IPR SUBV VF PERS 3 NBR SING SENS (6 5 4 3 2 1))))))
;;;          : (?Z NIL "PHVB" (K PHVB FS COORD RL ID CAT V ENONCP DECL VOIX ACT MT IPR
;;;          :  PHASE NONACC SUBV VF PERS 3 LINKS OUI NBR SING VALET Q ARGS A1))
;;;          : (?P1 (("la" "LE" (FS DES CAT D GNR FEM NBR SING))) (("boule" "BOULE"
;;;          :  (FS GOV CAT N SUBN NC GNR FEM NBR SING SENS 1))))
;;;          : (?Y NIL "DGN" (K GN FS OBJ RL ARG1 CAT (D N) SUBN NC GNR FEM PERS 3
;;;          :  NBR SING VALET N))
;;;          : (?W ((NIL "GN" (K GN FS SUJ RL ARG0 CAT N SUBN NP GNR MAS PERS 3
;;;          :  NBR SING VALET N)) (("Pierre" "*PIERRE" (FS GOV CAT N SUBN NP GNR MAS
;;;          :  PERS 3 NBR SING SENS 1)))) ((NIL "NV" (K PHVB CAT V ENONCP DECL
;;;          :  VOIX ACT MT IPR PHASE NONACC SUBV VF PERS 3 LINKS OUI RECHTS OUI
;;;          :  VCOOR OUI NBR SING VALET Q ARGS (A1 A0))) (("prend" "PRENDRE" (FS GOV
;;;          :  CAT V MT IPR SUBV VF PERS 3 NBR SING SENS (4 2 1))))))
;;;          : (?X NIL "PHVB" (K PHVB CAT V ENONCP DECL VOIX ACT MT IPR PHASE NONACC
;;;          :  SUBV VF PERS 3 LINKS OUI RECHTS OUI VCOOR OUI NBR SING VALET Q
;;;          :  ARGS (A1 A0)))
;;;          : )
;;;----------------------------------------------------------------------------
```

6.1.3. Patterns defined for the LIDIA-1.0 mockup

Let's see the patterns we have defined for the LIDIA-1 mockup.

i. Verbal coordination.

Pattern 1

```
(setf (get 'lidia-patterns 'cv-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'CS 'PHVB)
         (?* ?p2))))
```

and

Pattern 2

```
(setf (get 'lidia-patterns 'cv-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COORD)
          (?* ?p2)))))
```

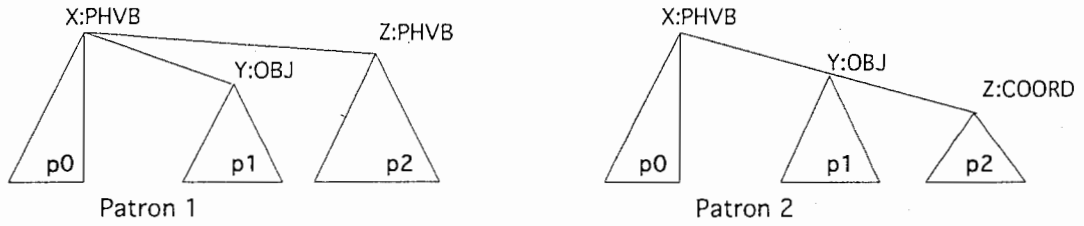


Figure 6.1: Patterns for verbal coordination

ii. Argument structure of the verb

Type 1: Il parle depuis l'école de cuisine.

Pattern 3

```
(setf (get 'lidia-patterns 'sav-t1-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'CIRC)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'FS 'OBJ)
         (?* ?p2))))
```

and

Pattern 4

```
(setf (get 'lidia-patterns 'sav-t1-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'CIRC)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)))))
```

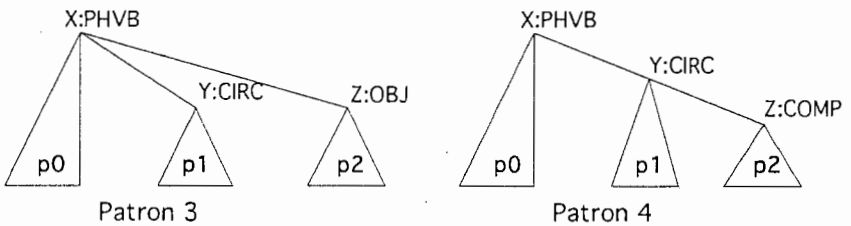


Figure 6.2: Patterns for the argument structure of the verb – type 1

Type 2: Il parle de l'école de cuisine.

Pattern 5

```
(setf (get 'lidia-patterns 'sav-t2-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'CIRC)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'FS 'OBJ)
         (?* ?p2))))
```

and

Pattern 6

```
(setf (get 'lidia-patterns 'sav-t2-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'CIRC)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)))))
```

and

Pattern 7

```
(setf (get 'lidia-patterns 'sav-t2-3)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)))))
```

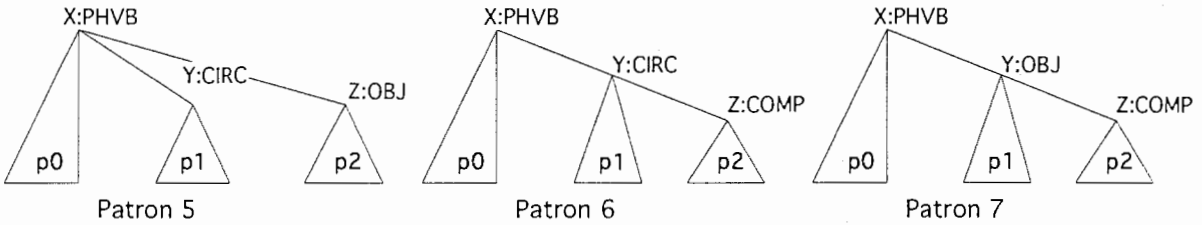


Figure 6.3: Patterns for the argument structure of the verb – type 2

Type 3: Marie voit l'homme dans le parc. Le capitaine a rapporteé un vase de chine.

Pattern 8

```
(setf (get 'lidia-patterns 'sav-t3-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'FS 'CIRC)
         (?* ?p2))))
```

and

Pattern 9

```
(setf (get 'lidia-patterns 'sav-t3-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)))))
```

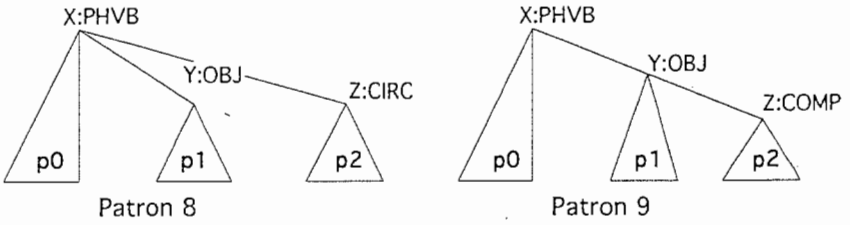


Figure 6.4: Patterns for the argument structure of the verb – type 3

Type 4: Marie voit l'homme dans (complement) le parc avec un télescope.

Pattern 10

```
(setf (get 'lidia-patterns 'sav-t4-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)))
        ((?is ?t node-prop-equal-p 'FS 'COMP)
         (?* ?p3))))
```

and

Pattern 11

```
(setf (get 'lidia-patterns 'sav-t4-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2))
         ((?is ?t node-prop-equal-p 'FS 'COMP)
          (?* ?p3)))))
```

and

Pattern 12

```
(setf (get 'lidia-patterns 'sav-t4-3)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)
          ((?is ?t node-prop-equal-p 'FS 'COMP)
           (?* ?p3))))))
```

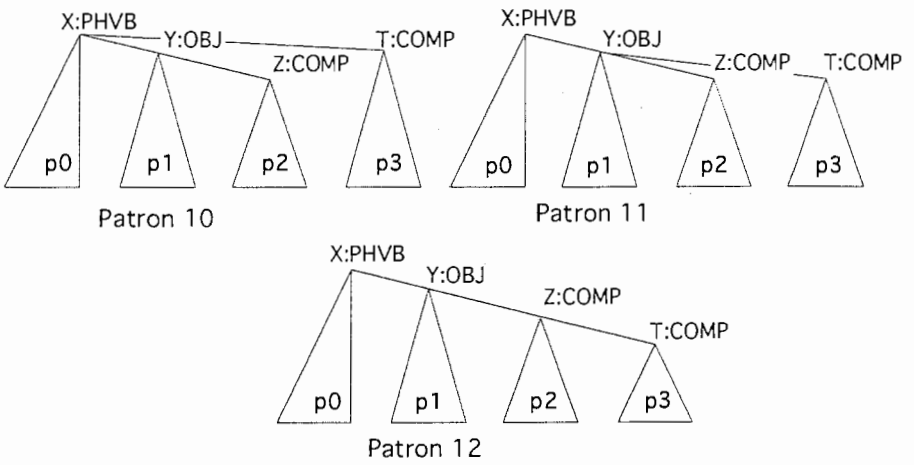


Figure 6.5: Patterns for the argument structure of the verb – type 4

Type 5: Marie voit l'homme dans (adverbial clause) le parc avec un télescope.

Pattern 13

```
(setf (get 'lidia-patterns 'sav-t5-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'FS 'CIRC
         (?* ?p2))
        ((?is ?t node-prop-equal-p 'FS 'COMP)
         (?* ?p3))))
```

and

Pattern 14

```
(setf (get 'lidia-patterns 'sav-t5-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'FS 'CIRC)
         (?* ?p2)
         ((?is ?t node-prop-equal-p 'FS 'COMP)
          (?* ?p3)))))
```

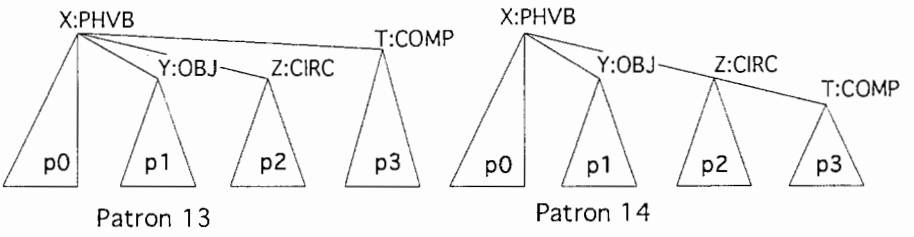


Figure 6.6: Patterns for the argument structure of the verb – type 5

Type 6: Le magistrat juge les enfants coupables.

Pattern 15

```
(setf (get 'lidia-patterns 'sav-t3-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'FS 'ATOBJ)
         (?* ?p2))))
```

and

Pattern 16

```
(setf (get 'lidia-patterns 'sav-t3-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)))))
```

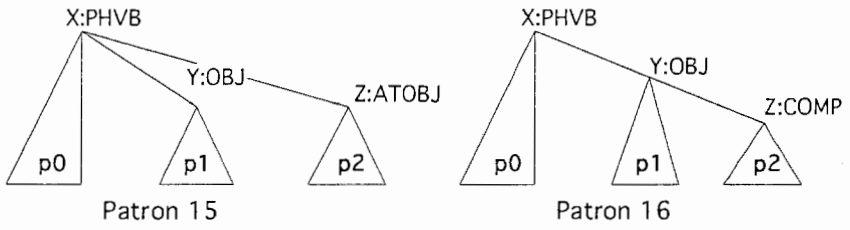


Figure 6.7: Patterns for the argument structure of the verb – type 6

Note

For other problems of verb argument structure, as in the sentences "Il parle de l'école de cuisine lyonnaise" or "Il écrit sur le pont du navire," the ambiguities will be solved in two steps. In the first step, the last phrase of the sentence will be removed so that the system will solve a labeling ambiguity. In the second step, the system will solve an ambiguity of subordination.

Cf. example 6.3.3.i. type2 page 59.

iii. Noun coordination

Type 1: On étudie l'évolution de la bourse et des investissements.

Pattern 17

```
(setf (get 'lidia-patterns 'nc-t1-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2))
         ((?is ?t node-prop-equal-p 'FS 'COORD)
          (?* ?p3)))))
```

and

Pattern 18

```
(setf (get 'lidia-patterns 'nc-t1-2)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)
          ((?is ?t node-prop-equal-p 'FS 'COORD)
           (?* ?p3))))))
```

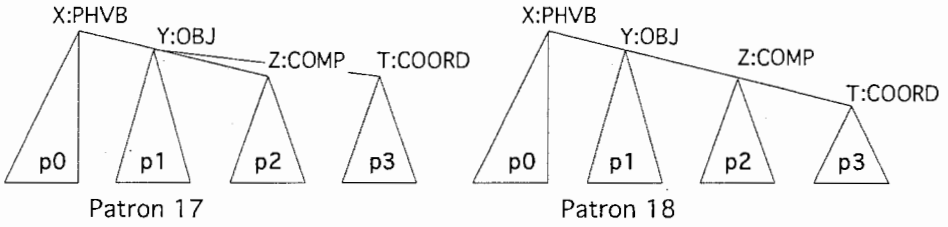


Figure 6.8: Patterns for noun coordination – type 1

Type 2: On étudie l'évolution de la structure du réseau et des investissements.

Pattern 19

```
(setf (get 'lidia-patterns 'nc-t2-1)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)
          ((?is ?t node-prop-equal-p 'FS 'COMP)
           (?* ?p3)))
        ((?is ?u node-prop-equal-p 'FS 'COORD)
         (?* ?p4))))))
```

and

Pattern 20

```
(setf (get 'lidia-patterns 'nc-t2-3)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)
          ((?is ?t node-prop-equal-p 'FS 'COMP)
           (?* ?p3))
          ((?is ?u node-prop-equal-p 'FS 'COORD)
           (?* ?p4)))))))
```

and

Pattern 21

```
(setf (get 'lidia-patterns 'nc-t2-3)
      '((?is ?x node-prop-equal-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'OBJ)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)
          ((?is ?t node-prop-equal-p 'FS 'COMP)
           (?* ?p3)
           ((?is ?u node-prop-equal-p 'FS 'COORD)
            (?* ?p4)))))))
```

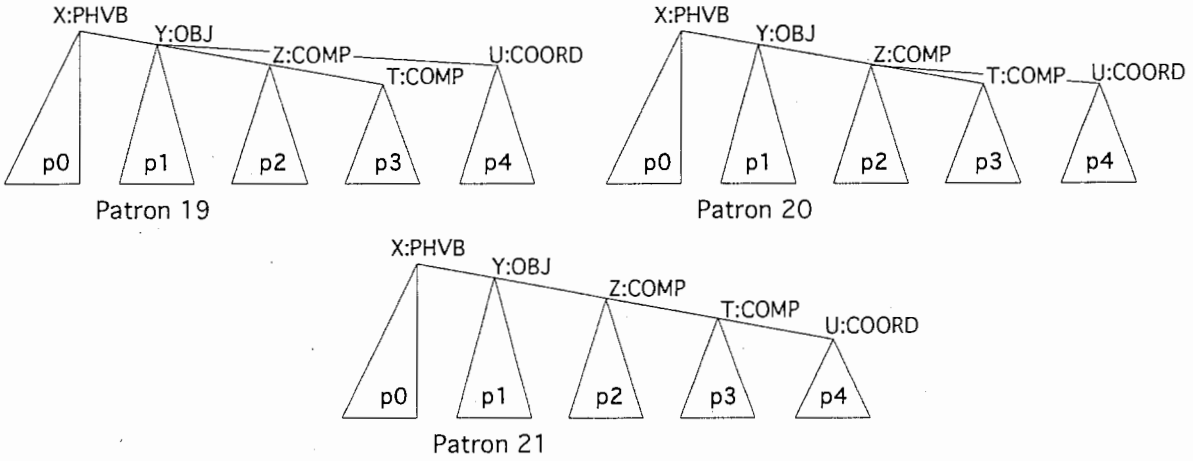


Figure 6.9: Patterns for noun coordination – type 2

Type 3: L'évolution de la structure du reseau et des investissements est étudiée.

Pattern 22

```
(setf (get 'lidia-patterns 'nc-t3-1)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COMP)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2))
         ((?is ?t node-prop-equal-p 'FS 'COORD)
          (?* ?p3)))))
```

and

Pattern 23

```
(setf (get 'lidia-patterns 'nc-t3-2)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COMP)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COMP)
          (?* ?p2)
          ((?is ?t node-prop-equal-p 'FS 'COORD)
           (?* ?p3))))))
```

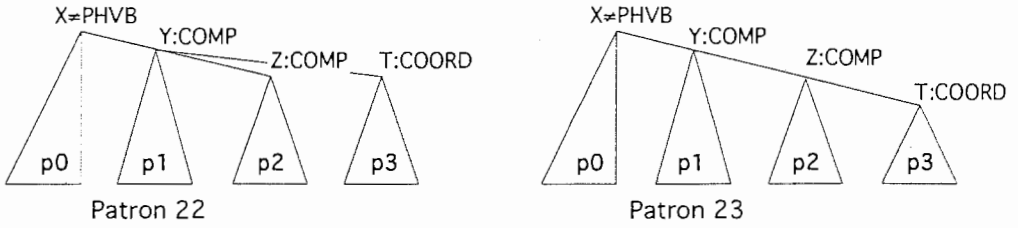


Figure 6.10: Patterns for noun coordination – type 3

For more complex coordinations, some other patterns will be necessary.

iv. Adjective coordination

Type 1: Il prend des cahiers et des classeurs noirs.

Pattern 24

```
(setf (get 'lidia-patterns 'ca-t1-1)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COORD)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'CS 'GADJ)
          (?* ?p2)))))
```

and

Pattern 25

```
(setf (get 'lidia-patterns 'ca-t1-2)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COORD)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'CS 'GADJ)
          (?* ?p2))))
```

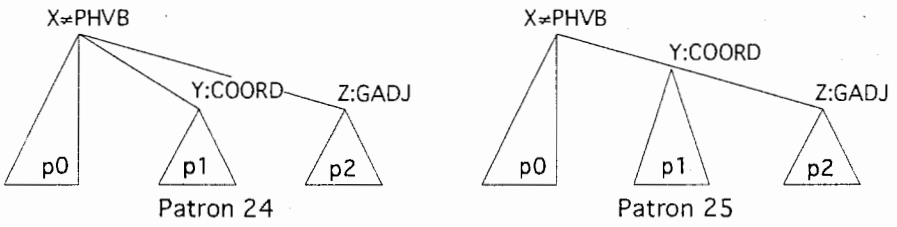


Figure 6.11: Patterns for adjectival coordination – type I

For more complex problems as in the following sentence: "Il prend des cahiers, des feutres, des stylos et des classeurs noirs.", some other patterns will be necessary as defined next page. The other way we could choose to handle that kind of problem is to have the pattern description language to handle regular expressions, something like "pronoun verb (coord+) adjective".

Type 2 : Il prend des cahiers, des stylos et des classeurs noirs.

Pattern 26

```
(setf (get 'lidia-patterns 'ca-t2-1)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COORD)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COORD)
          (?* ?p2)))
        ((?is ?t node-prop-equal-p 'FS 'GADJ)
         (?* ?p3))))
```

and

Pattern 27

```
(setf (get 'lidia-patterns 'ca-t2-2)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COORD)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COORD)
          (?* ?p2))
         ((?is ?t node-prop-equal-p 'FS 'GADJ)
          (?* ?p3)))))
```

and

Pattern 28

```
(setf (get 'lidia-patterns 'ca-t2-3)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COORD)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'FS 'COORD)
          (?* ?p2)
          ((?is ?t node-prop-equal-p 'FS 'GADJ)
           (?* ?p3))))))
```

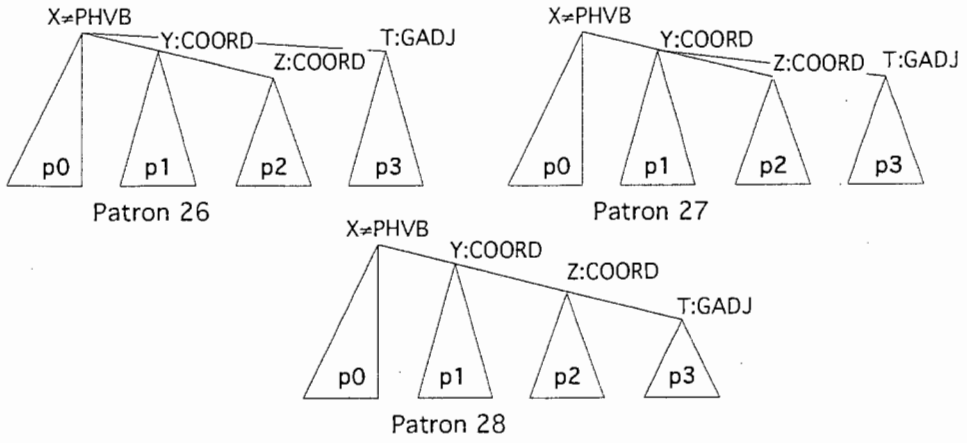


Figure 6.12: Patterns for adjectival coordination – type 2

v. **Subordination**

Pattern 29

```
(setf (get 'lidia-patterns 'sub-t1-1)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COMP)
         (?* ?p1)
         ((?is ?z node-prop-equal-p 'CS 'ATG)
          (?* ?p2)))))
```

and

Pattern 30

```
(setf (get 'lidia-patterns 'sub-t1-2)
      '((?is ?x node-prop-diff-p 'CS 'PHVB)
        (?* ?p0)
        ((?is ?y node-prop-equal-p 'FS 'COMP)
         (?* ?p1))
        ((?is ?z node-prop-equal-p 'CS 'ATG)
         (?* ?p2))))
```

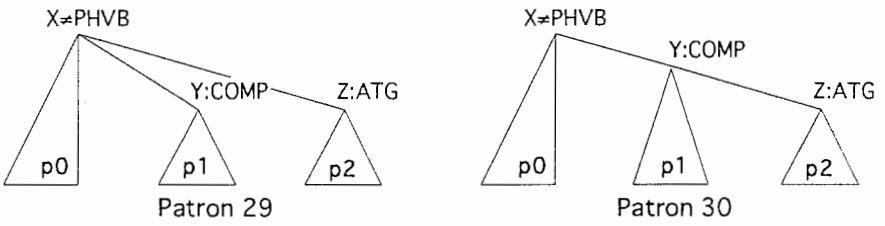


Figure 6.13: Patterns for subordination

We can make the same remark as before about the treatment of more general ambiguities.

6.1.4. Important remark

If the matching process does not work, the last phrase of the sentence is removed from all the parse trees and the process starts again.

For example, for the sentences "Il parle de l'école de cuisine lyonnaise" the matching process will give no result. Thus the process will start again on the mmc-structure restricted to "Il parle de l'école de cuisine". Then the disambiguation process will recognize an ambiguity of the argument structure of the verb "parler". In the second step, an amnbiguity of subordination will be recognized (cf. examples § 6.3.3.i. type 2 and §6.3.3.i type 3,4,5: second sentence).

6.2. Dialogue item production method

For the ambiguities described with sets of patterns, we can now describe the dialogue item production method associated with each pattern. For the other ambiguities, we will describe how the dialogue is produced.

6.2.1. Lexical ambiguity

We examine the value of the variable sense of each leaf in the chosen solution in the mmc-structure. If the variable has a set of values, they are the word senses the analyzer has not been able to reject.

The system produces a dialogue where each item is a possible sense. The senses are found in the lexical database.

6.2.2. Categorial ambiguity

i. Verbal coordination

Verbal coordination implies patterns 1 and 2.

Pattern 1	Subject(?p0) NoyauVerbal(?p2) Text(?p1)
Pattern 2	Subject(?p0) NoyauVerbal(?p0) But_Coord(?p2)

ii. Non-verbal coordination

The system produces a dialogue containing one item for each different labeling. The dialogue items consist of the words of the original sentence with the syntactic class associated with each ambiguous one.

6.2.3. Geometrical ambiguity

i. Argument structure of the verb

Type 1

Pattern 3	Text(?p0) Text(?p1) Substitute(Subordinator(?p2) #Objet) But_Sub(?p2)
Pattern 4	Text(?p0) Subordinator(?p1) Bracket(But_Sub(?p1) Text(?p2))

Type 2

Pattern 5	Text(?p0) Substitute(Subordinator(?p1) #Circ) But_Sub(?p1)
	Substitute(Subordinator(?p2) #Objet) But_Sub(?p2)

Pattern 6	Text(?p0) Substitute(Subordinator(?p1) #Circ) But_Sub(?p1) Text(?p2))
Pattern 7	Text(?p0) Substitute(Subordinator(?p1) #Obj) But_Sub(?p1) Text(?p2))

Type 3

Pattern 8	Text(?p2) , Text(?p0) Text(?p1)
Pattern 9	Text(?p0) Determiner(?p1) Bracket(But_Det(?p1), Text(?p2))

Type 4

Pattern 10	Text(?p3) , Text(?p0) Determiner(?p1) Bracket(But_Det(?p1), Text(?p2))
Pattern 11	Text(?p0) Determiner(?p1) Bracket(Bracket(But_Det(?p1), Text(?p3)), Text(?p2))
Pattern 12	Text(?p0) Determiner(?p1) Bracket(But_Det(?p1), Reg(?p2), Bracket(But_Sub(?p2), Text(?p3)))

Type 5

Pattern 13	Text(?p3) , Text(?p2) , Text(?p0)) Text(?p1))
Pattern 14	Bracket(Text(?p2) , Text(?p3)) Text(?p0)) Text(?p1))

Type 6

Pattern 15	Text(?p0) Text(?p2)) Text(?p1)
Pattern 16	Text(?p0) Determiner(?p1) Bracket(But_Det(?p1), Text(?p2)))

ii. Noun coordination

Type 1

Pattern 17	Text(?p0) But_coord(?p3)
Pattern 18	Text(?p0) Text(?p1) But_coord(?p3)

Type 2

Pattern 19	Text(?p0) But_coord(?p4)
Pattern 20	Text(?p0) Text(?p1) But_coord(?p4)
Pattern 21	Text(?p0) Text(?p1) Text(?p2) But_coord(?p4)

Type 3

Pattern 22	Text(?p0) But_coord(?p3)
Pattern 23	Text(?p0) Text(?p1) But_coord(?p3)

iii. Adjective coordination

Type 1

Pattern 24	Distribute(Text(?p0), Text(?p1), But_Coord(?p2), Coord(?p2), (1, 3), (2, 3))
Pattern 25	Distribute(Text(?p0), Text(?p1), But_Coord(?p2), Coord(?p2), (1), (2, 3))

Type 2

Pattern 26	Distribute(Text(?p0), Text(?p1), Text(?p2), But_Coord(?p3), Coord(?p3), (1, 4), (2, 4), (3, 4))
Pattern 27	Distribute(Text(?p0), Text(?p1), Text(?p2), But_Coord(?p3), Coord(?p3), (1), (2, 4), (3, 4))
Pattern 28	Distribute(Text(?p0), Text(?p1), Text(?p2), But_Coord(?p3), Coord(?p3), (1), (2), (3, 4))

iv. Subordination

Pattern 29	Determiner(?p0) Bracket(But_Det(?p0), Text(?p3)) Text(?p2)
Pattern 30	But_Sub(?p0) , Reg(?p1) Bracket(But_Sub(?p1), Text(?p2))

6.2.4. Labeling ambiguity

As has been said before, in the LIDIA-1 mockup labeling ambiguity is not solved with the pattern matching technique. The system projects the relevant information of the trees and inspect the produced lists to detect the ambiguity.

i. Logico-semantic and syntactic labeling

The system replaces the ambiguous prepositions by non-ambiguous ones. The new preposition is chosen according to the information found in the lexical database.

ii. Argument order for a direct transitive verb

When ambiguous phrases are detected, the standardized sentences are produced.

6.3. Results

6.3.1. Lexical ambiguity

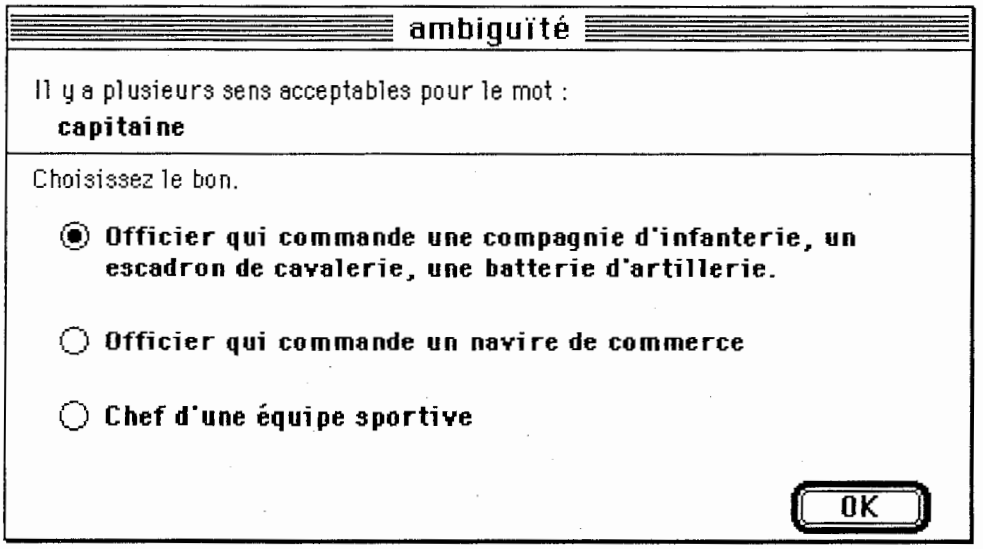


Figure 6.14: Dialogue for a word sense ambiguity – example 2[65]

[65] For the noun "capitaine", the possible word senses are: ① officer commanding a military unit, ② officer commanding a trade boat, ③ head of a sports team.

6.3.2. Categorial ambiguity

i. Verbal coordination

The sentence	Il atteint la grange et la ferme. (cf. footnote 39)
The matching:	– p0 = Il atteint – p1 = la grange – p2 = et la ferme

The produced dialogue:

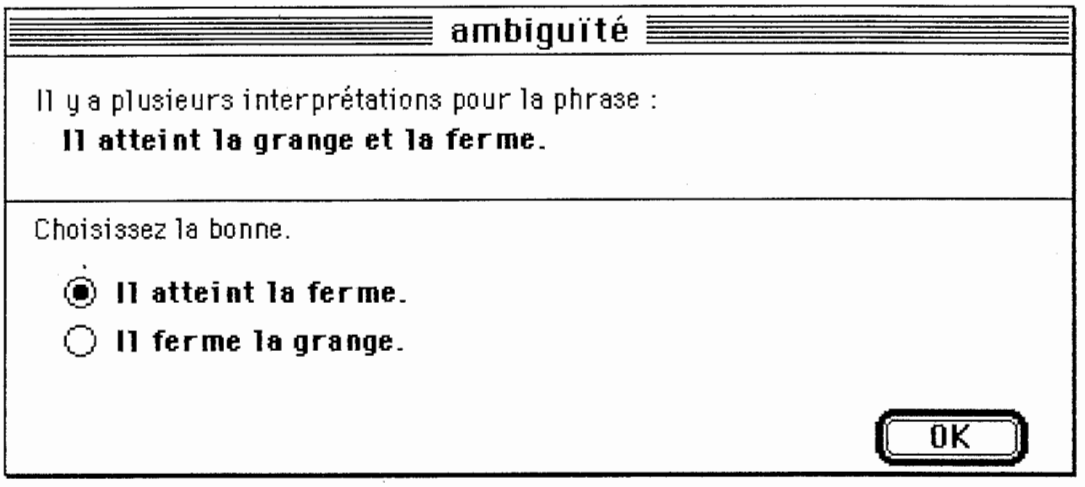


Figure 6.15: Dialogue for a verbal coordination ambiguity[66]

ii. Non-verbal coordination

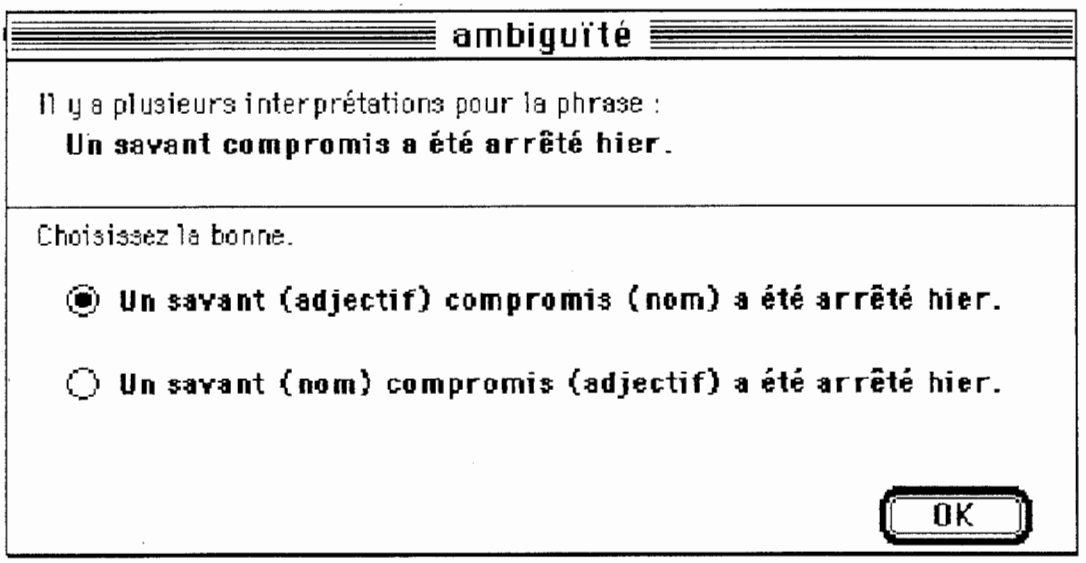


Figure 6.16: Dialogue for a simple categorial ambiguity – example 1

[66] The proposed interpretations are: ① He reaches the farm, ② He closes the barn.

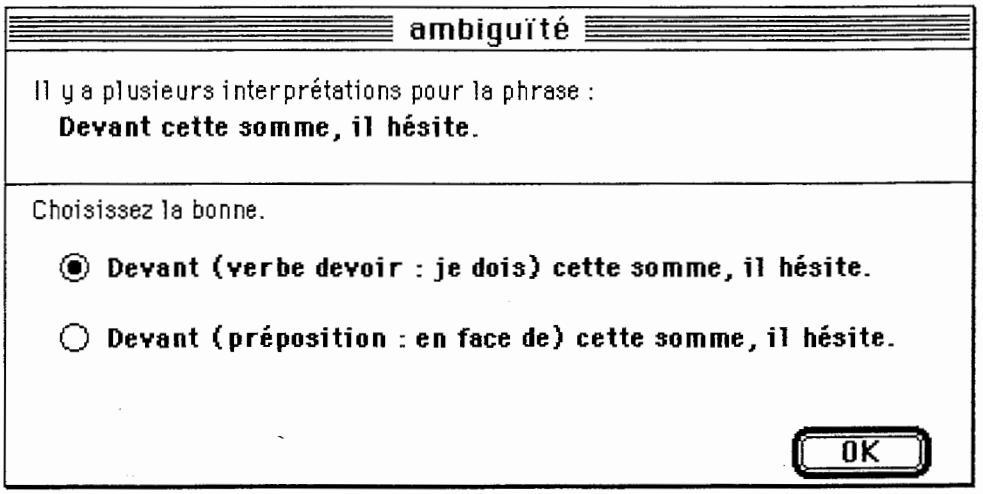


Figure 6.17: Dialogue for a simple categorial ambiguity – example 2

Pour la préposition, on propose un synonyme, et pour le verbe on propose sa conjugaison à la première personne du présent de l'indicatif (pour les arbres, cf figure 9.2).

6.3.3. Geometrical ambiguity

i. Argument structure of the verb

Type 1

The sentence	Il parle depuis l'école de cuisine. (cf. footnote 19 & 20)
The patterns	3 4
The matching:	– p0 = Il parle – p1 = depuis l'école – p2 = de cuisine

The produced dialogue:

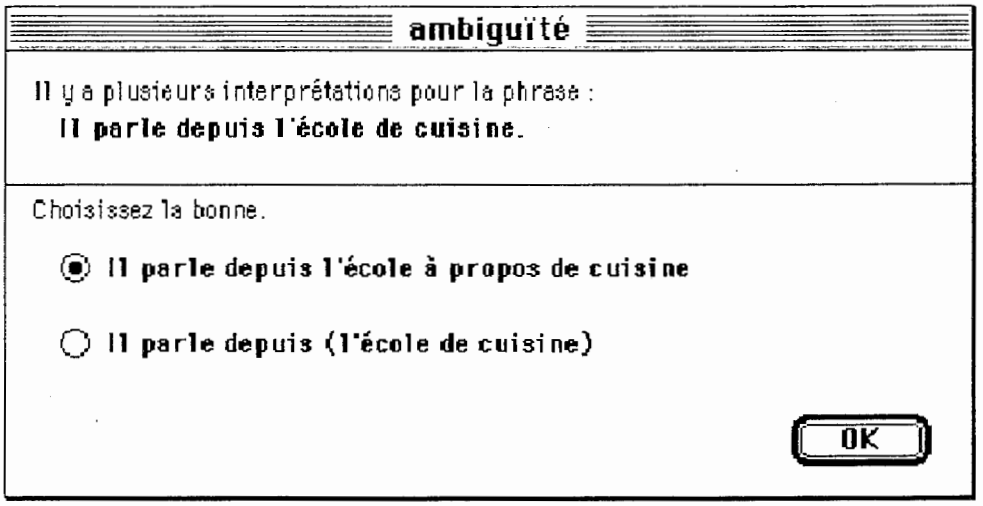


Figure 6.18: Dialogue for an ambiguity of argument structure of the verb, type 1 [67]

[67] The proposed interpretations are: ① He speaks from the school about cooking, ② He speaks from (the coocking school).

Type 2

The sentence	Il parle de l'école de cuisine lyonnaise. (step 1) (cf. footnote 17 & 45)
The patterns	5 6 7
The matching	– p0 = Il parle – p1 = de l'école – p2 = de cuisine

The produced dialogue:

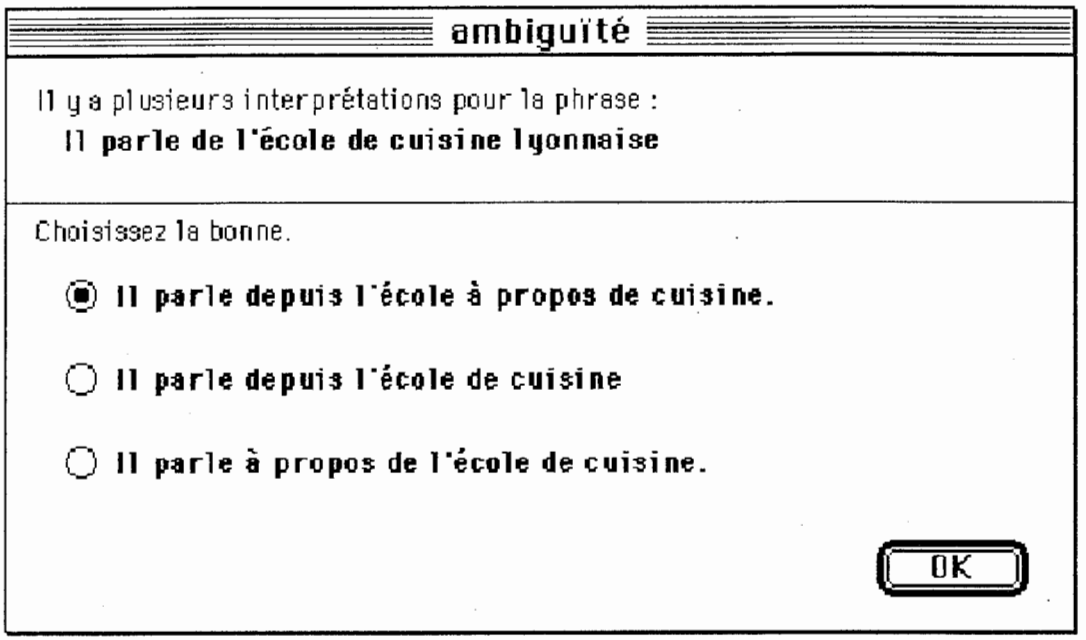


Figure 6.19: Dialogue for an ambiguity of argument structure of the verb, type 2[68]

The sentence	Il parle de l'école de cuisine lyonnaise. (step 2)
The patterns	29 30
The matching	– p0 = de l'école – p1 = de cuisine – p2 = lyonnaise

The produced dialogue:

[68] The proposed interpretations are: ① He speaks from the school about cooking, ② He speaks from the cooking school, ③ He speaks about the cooking school.

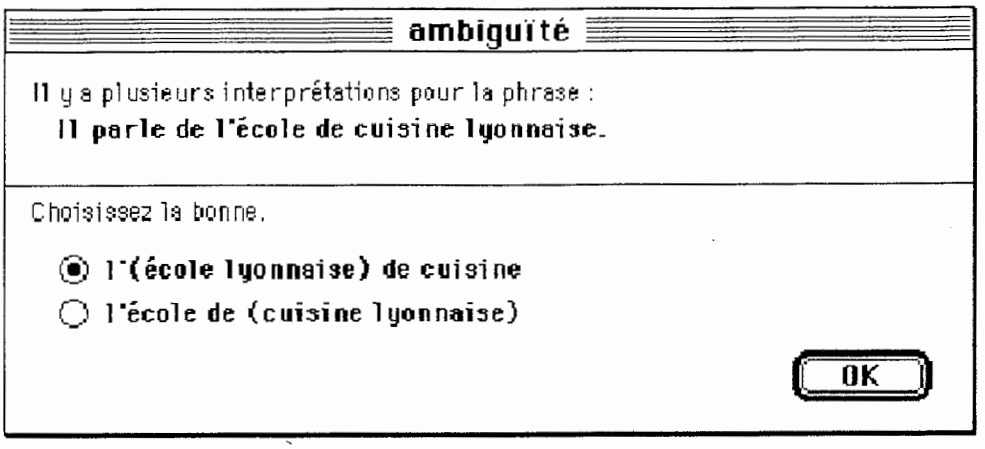


Figure 6.20:Dialogue for a subordination ambiguity[69]

Type 3, type 4, type 5

The sentence	Le capitaine a rapporté un vase de chine. (cf. footnote 7)
The patterns	8 9
The matching	– p0 = Le capitaine a rapporté – p1 = un vase – p2 = de chine

The produced dialogue:

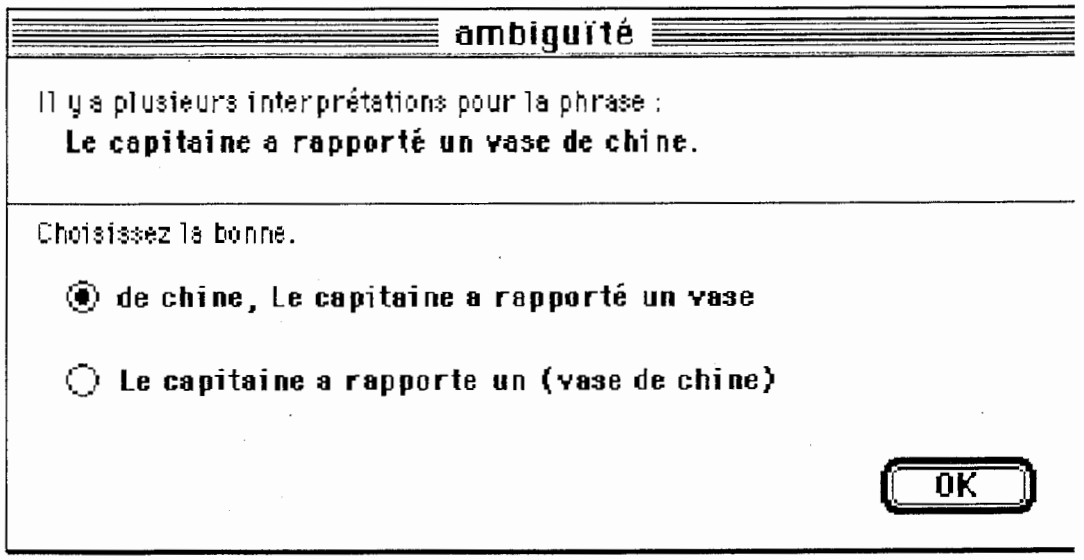


Figure 6.21: Dialogue for an ambiguity of argument structure of the verb, type 3 [70]

The sentence	Marie voit l'homme dans la parce avec un télescope. (step 1)
The patterns	8 9

[69] The proposed interpretations are: ① the (lyonnaise school) of cooking, ② the school of (lyonnaise coocking).
[70] The proposed interpretations are: ① from China, the capitain brought back a vase, ② The capitain brought back a (Chinese vase).

The matching	– p0 = Marie voit – p1 = l'homme – p2 = dans le parc

The produced dialogue:

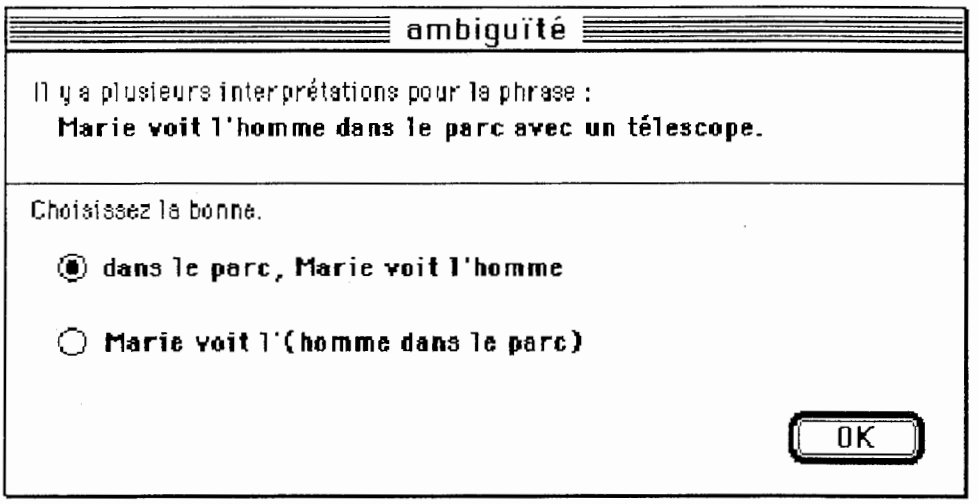


Figure 6.22: Dialogue for an ambiguity of argument structure of the verb, type 3[71]

The sentence	Marie voit l'homme dans la parc avec un télscope. (step 2.1)
The patterns	13 14
The matching	– p0 = Marie voit – p1 = l'homme – p2 = dans la parc – p3 = avec un télescope

The produced dialogue:

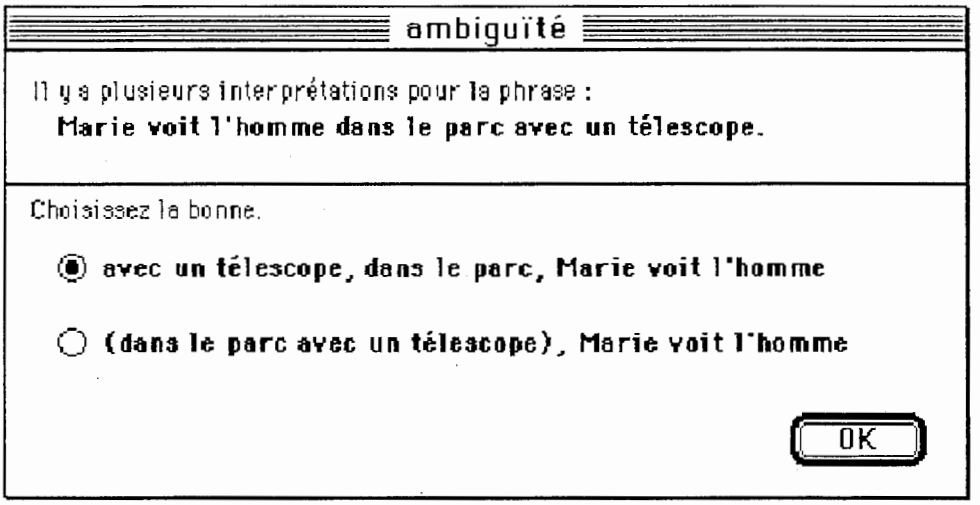


Figure 6.23: Dialogue for an ambiguity of argument structure of the verb ,type 4 [72]

[71] The proposed interpretations are: ① in the park, Marie sees the man, ② Marie sees the (man in the park).
[72] The proposed interpretations are: ① with a telescope, in the park, Marie sees the man, ② (in the park with a telescope), Marie sees the man.

The sentence	Marie voit l'homme dans le parc avec un télescope. (step 2.2)
The patterns	10 11 12
The matching	– p0 = Marie voit – p1 = l'homme – p2 = dans la parc – p3 = avec un télescope

The produced dialogue:

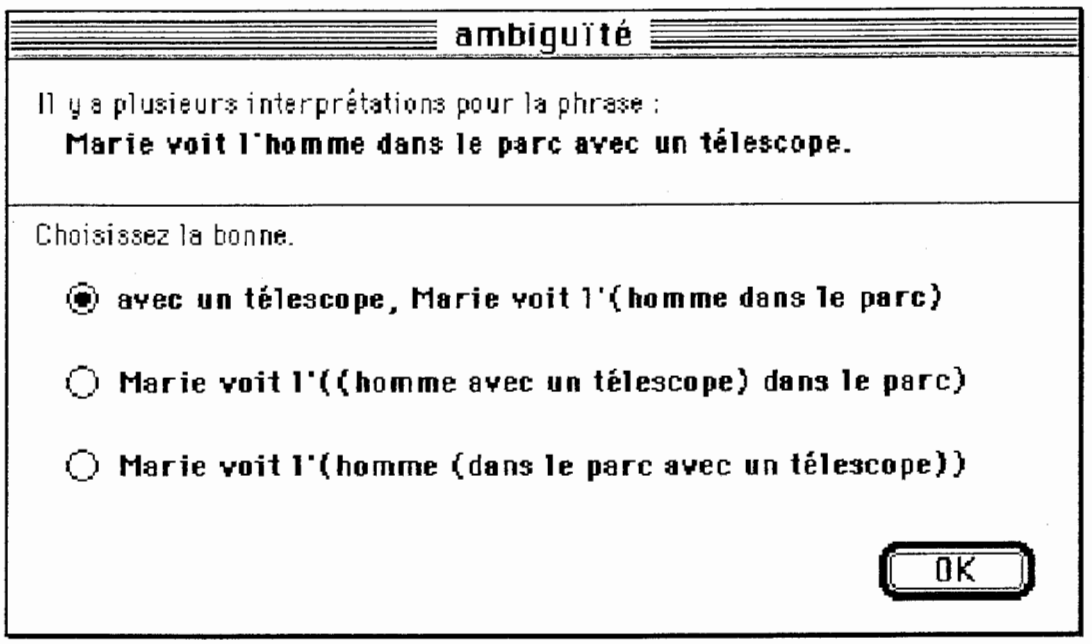


Figure 6.24: Dialogue for an ambiguity of argument structure of the verb, type 5[73]

Type 6

The sentence	Le magistrat juge les enfants coupables.
The patterns	15 16
The matching	– p0 = Le magistrat juge – p1 = les enfants – p2 = coupables

The produced dialogue:

[73] The proposed interpretations are: ① with a telescope, Marie sees the (man in the park), ② Marie sees the ((man with a telescope) in the park), ③ Marie sees the (man (in the park with a telescope)).

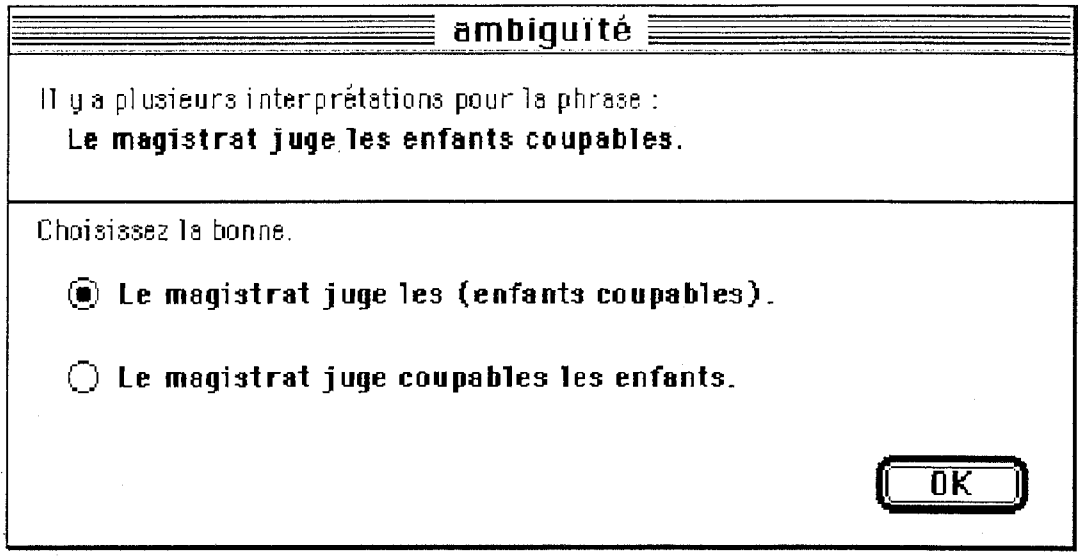


Figure 6.25: Dialogue for an ambiguity of argument structure of the verb type 6[74]

ii. Noun Coordination

Type 1

The sentence	On étudie l'évolution de la bourse et des investissements. (cf. footnote 47)
The patterns	17 18
The matching	– p0 = On étudie – p1 = l'évolution – p2 = de la bourse – p3 = et des investissements

The produced dialogue:

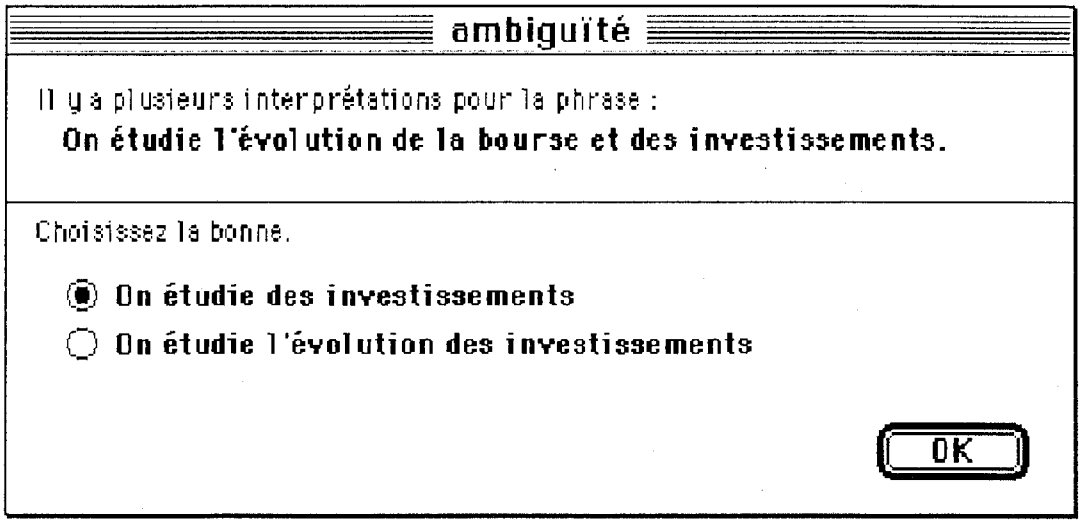


Figure 6.26: Dialogue for a noun coordination type 1 ambiguity[75]

[74] The proposed interpretations are: ① The magistrate judge the (guilty children), ② The magistrate found that the children were guilty.

Type 2

The sentence	On étudie l'évolution de la structure du réseau et des investissements .
The patterns	19 20 21
The matching	– p0 = On étudie – p1 = l'évolution – p2 = de la structure – p3 = du réseau – p4 = et des investissements

The produced dialogue:

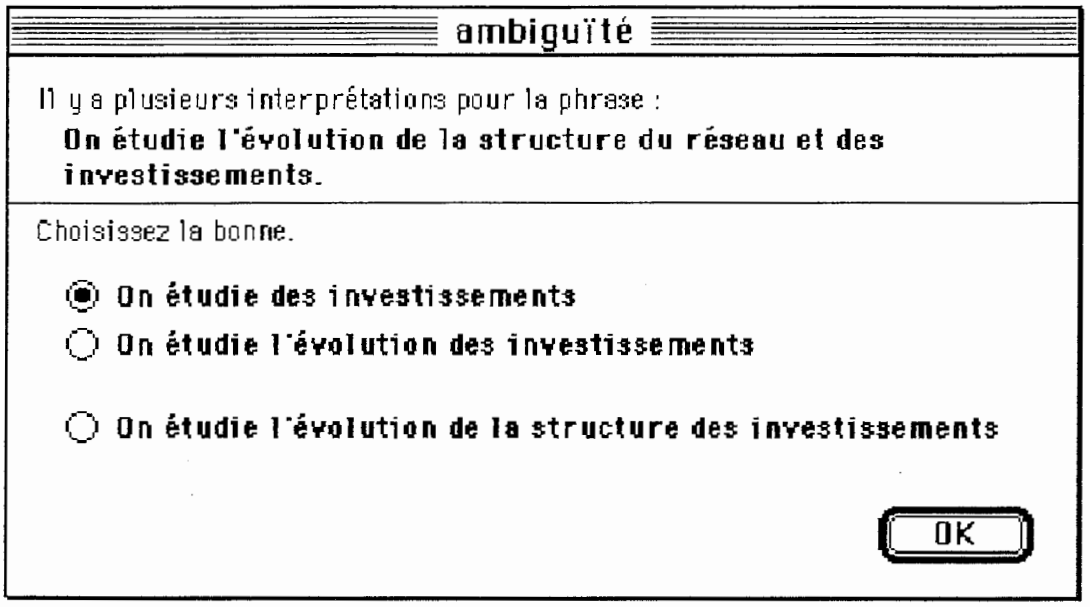


Figure 6.27: Dialogue for a noun coordination type 2 ambiguity[76]

Type 3

The sentence	L'évolution de la structure du réseau et des investissements est étudiée. (cf. footnote 21)
The patterns	22 23
The matching	– p0 = L'évolution – p1 = de la structure – p2 = du réseau – p3 = et des investissements

The produced dialogue:

[75] The proposed interpretations are: ① We study the investments, ② We study the evolution of the investments.
[76] The proposed interpretations are: ① We study the investments, ② We study the evolution of the investments, ③ We study the evolution of the structure of the investments.

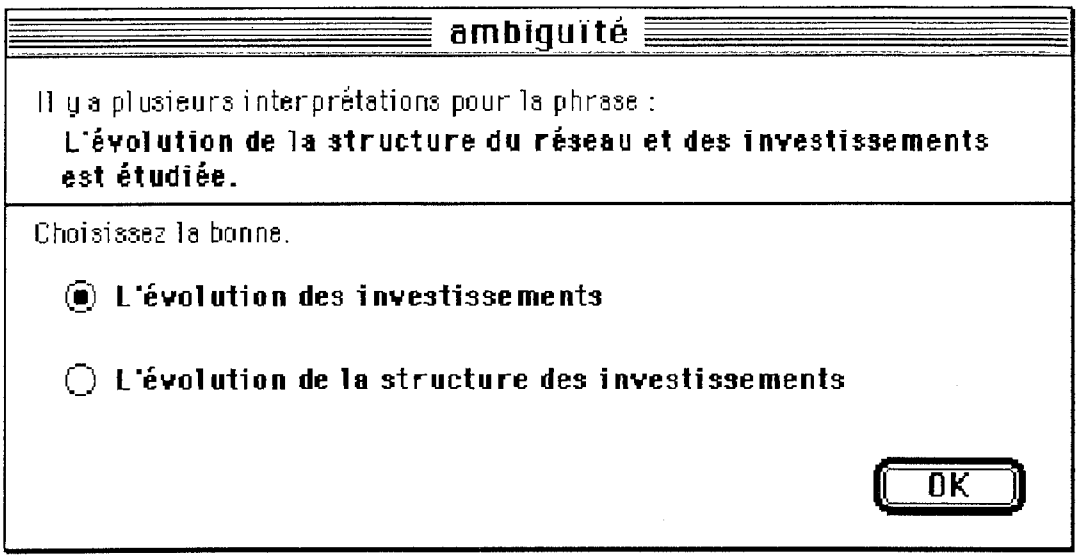


Figure 6.28: Dialogue for a noun coordination type 3 ambiguity[77]

iii. Adjective coordination

Type 1

The sentence	Il prend des cahiers et des classeurs noirs. (cf. footnote 24)
The patterns	24 25
The matching	– p0 = des cahiers – p1 = et des classeurs – p2 = noirs

The produced dialogue:

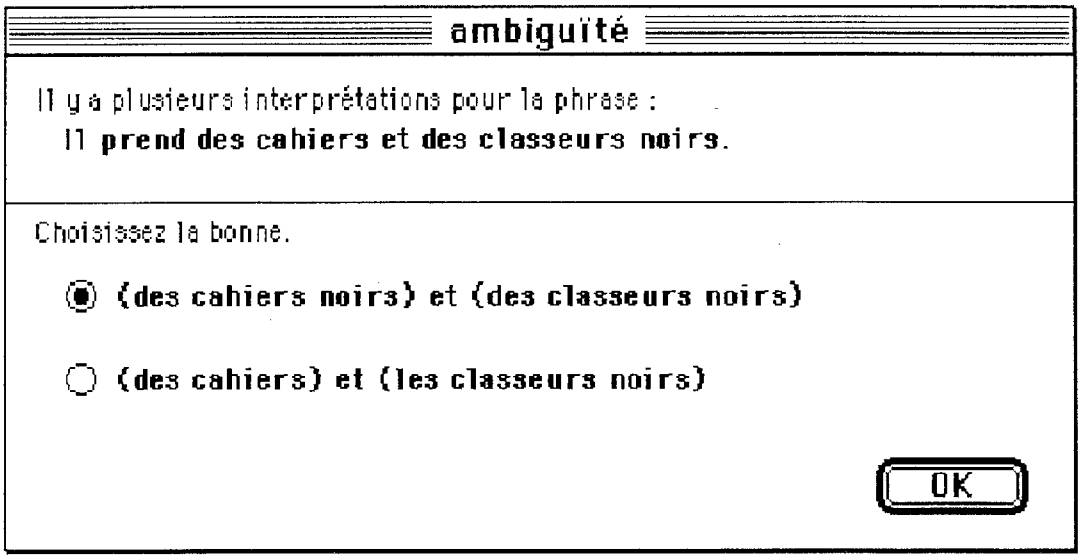


Figure 6.29: Dialogue for an adjective coordination type 1 ambiguity[78]

[77] The proposed interpretations are: ① The evolution of the investments, ② The evolution of the structure of the investments.

[78] The proposed interpretations are: ① (black notebooks) and (black files), ② (files) and (black notebooks).

Type 2

The sentence	Il prend des cahiers, des stylos et des classeurs noirs.
The patterns	26 27 28
The matching	– p0 = des cahiers – p1 = des stylos – p2 = et des classeurs

The produced dialogue:

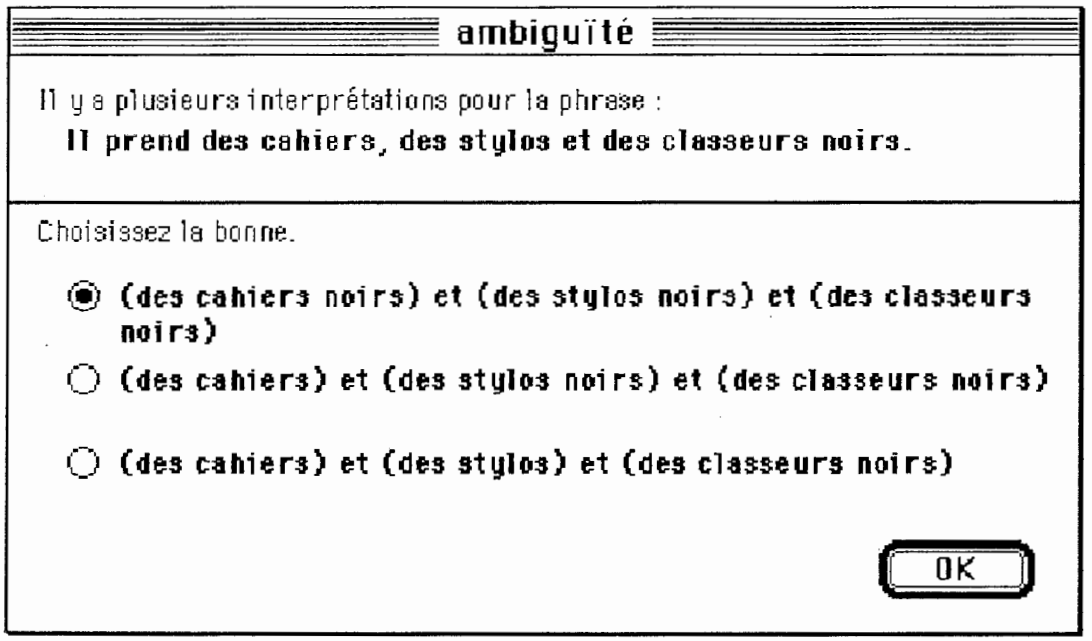


Figure 6.30: Dialogue for an adjective coordination type 2 ambiguity[79]

iv. Subordination

The sentence	Elle épouse un professeur de droit anglais. (cf. footnote 27)
The patterns	29 30
The matching	– p0 = un professeur – p1 = de droit – p2 = anglais

The produced dialogue:

[79] The proposed interpretations are: ① (black notebooks) and (black pens) and (black files), ② (notebooks) and (black pens) and (black files), ③ (notebooks) and (pens) and (black files).

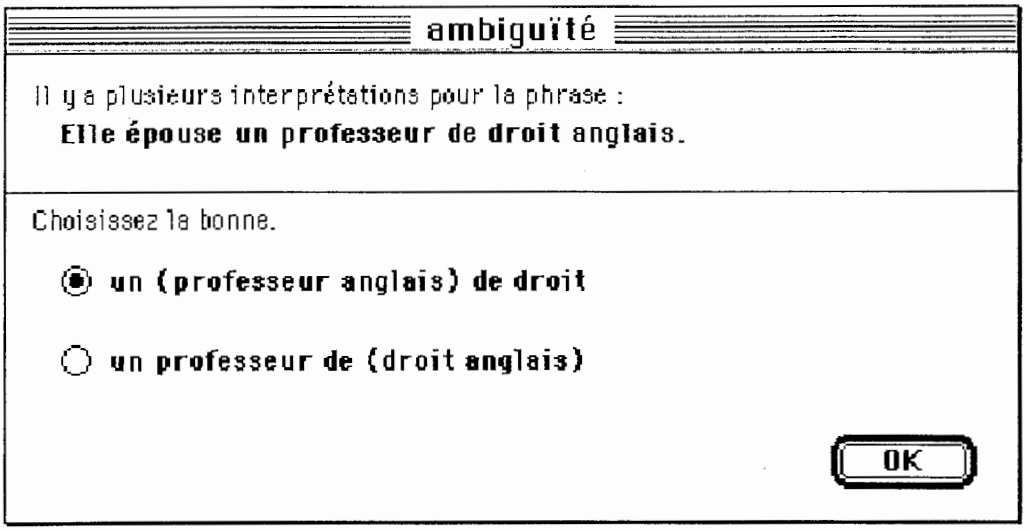


Figure 6.31: Dialogue for a subordination ambiguity[80]

The sentence	L'école de cuisine ouvre ses portes.
The patterns	29 30
The matching	– p0 = L'école – p1 = de cuisine – p2 = lyonnaise

The produced dialogue:

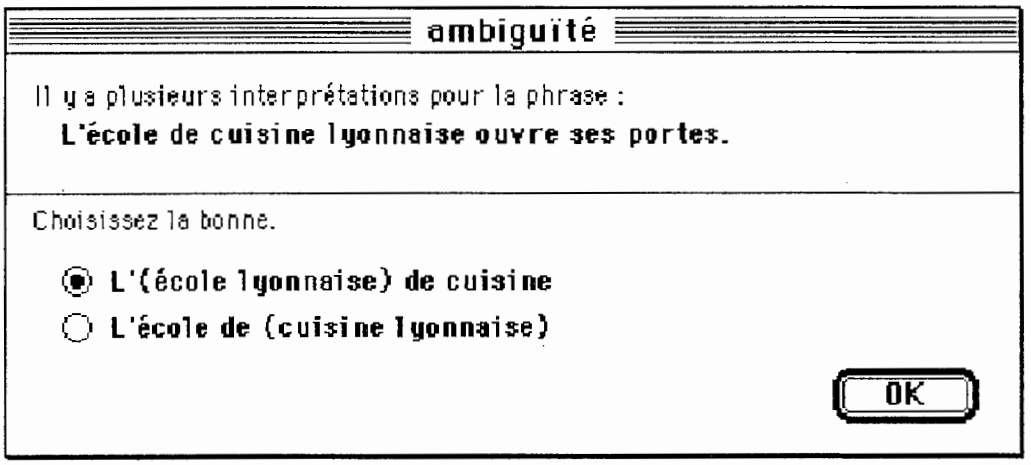


Figure 6.32: Dialogue for a subordination ambiguity[81]

6.3.4. Labeling ambiguity

i. Logico-sémantic labelling

The sentence	Pierre fait porter des chocolats à Lucie. (cf. footnote 30)

[80] The proposed interpretations are: ① an (English teacher) of law, ② a teacher of (English law).
[81] The proposed interpretations are: ① The (lyonnaise school) of cooking, ② The school of (lyonnaise cooking).

The produced dialogue:

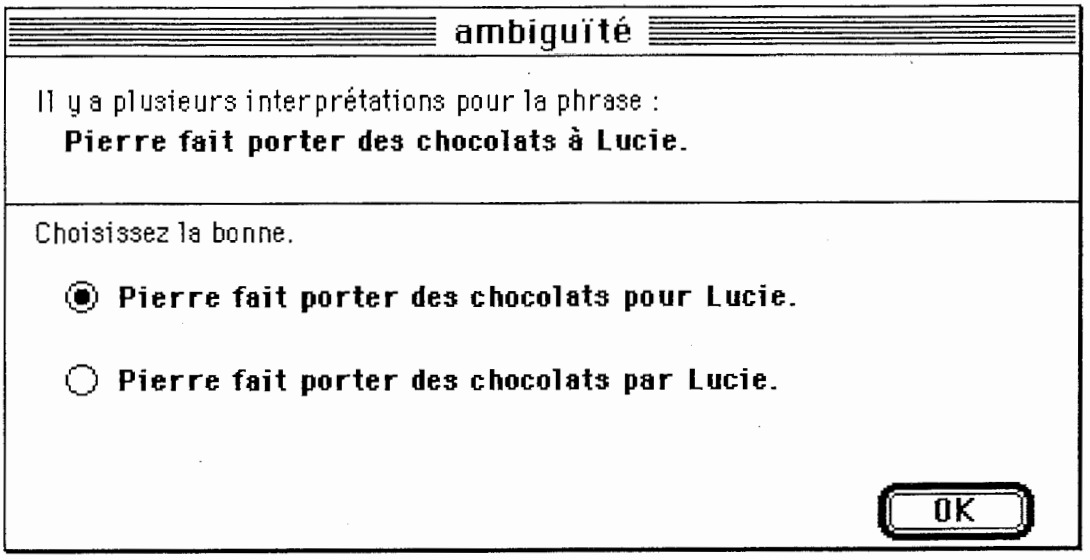


Figure 6.33: Dialogue for a logico-semantic labeling ambiguity[82]

ii. Syntactic labelling

The sentence	Je vous parle de la tour Eiffel. (cf. footnote 33)

The produced dialogue:

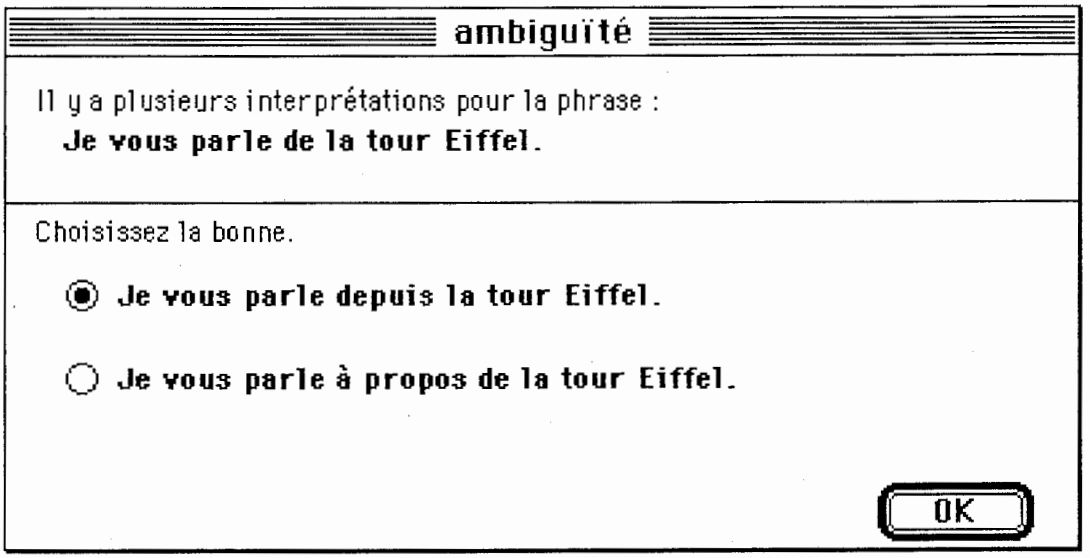


Figure 6.34: Dialogue for a syntactic labeling ambiguity – example 1[83]

The sentence	Je vote pour ma mère.

[82] The proposed interpretations are: ① Pierre is having chocolates sent for Lucie, ② Pierre is having chocolates sent by Lucie.

[83] The proposed interpretations are: ① I speak to you from the Eiffel Tower, ② I speak to you about the Eiffel Tower.

The produced dialogue:

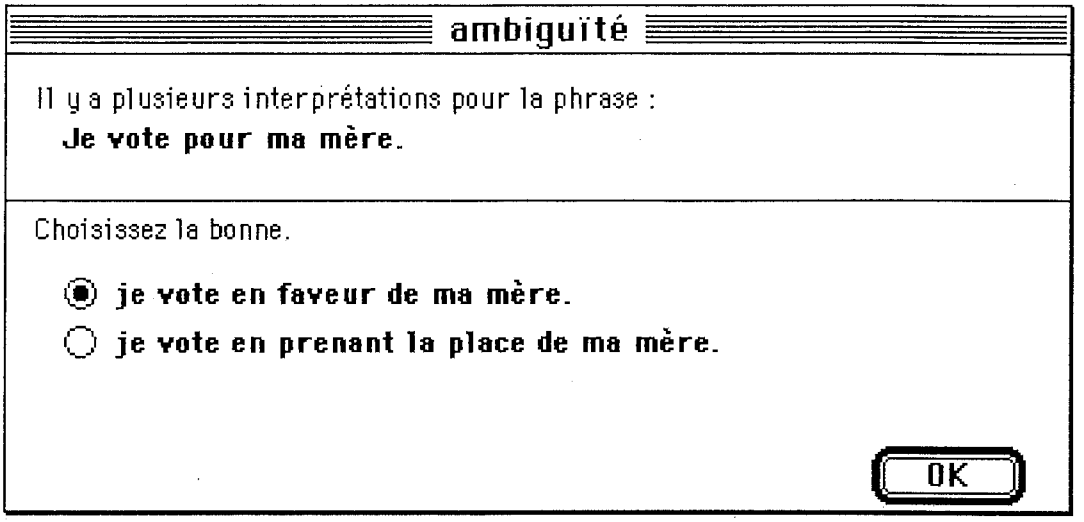


Figure 6.35: Dialogue for a syntactic labeling ambiguity – example 2[84]

iii. Argument order for a direct transitive verb

The sentence	Quel auteur cite ce conférencier ? (cf. footnote 36)

The produced dialogue:

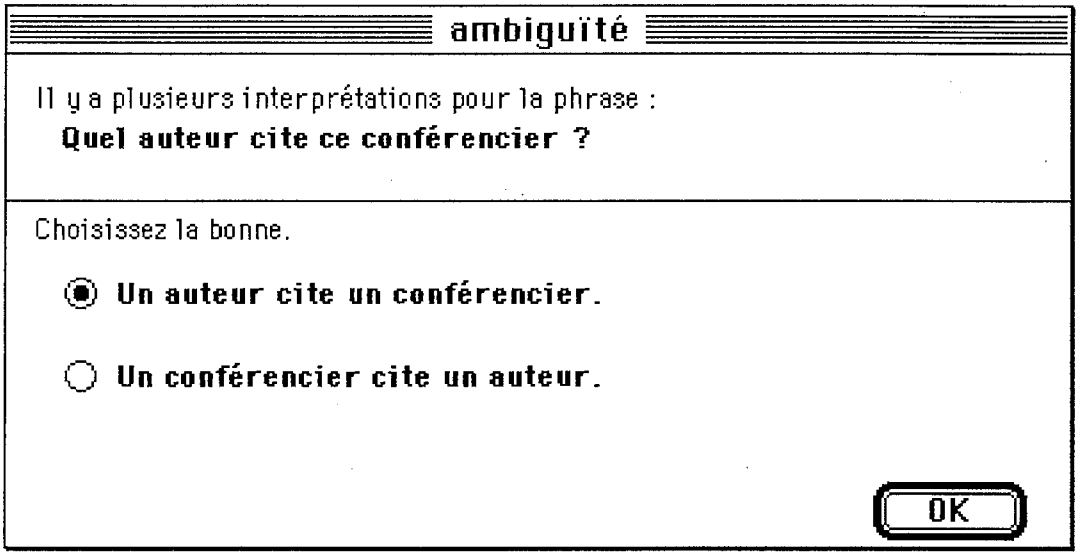


Figure 6.36: Dialogue for an ambiguity of argument order of a direct transitive verb[85]

[84] The proposed interpretations are: ① I vote in favor of my mother, ② I vote instead of my mother.
[85] The proposed interpretations are: ① An author quotes a lecturer, ② A lecturer quotes an author.

References

Ainsworth W. A. & Pratt S. R. (1992). *Feedback strategies for error correction in speech recognition systems.* in International Journal of Man-Machine Studies. vol. **36**(6) : pp. 833-842.

Béjoint H. (1990). *Dictionaries and the Acquisition of Polysemy.* Proc. The Lódz Session of the 1990 Maastricht-Lódz Duo Colloquium on "Translation and Meaning". Lódz, Poland. 20-22 septembre 1990, vol. : pp. 307-312.

Blanchon H. (1990). *HyperAriane : architecture de la partie "micro" d'un système de TAO personnelle.* Rap. GETA, IMAG, UJF&INPG. Rapport de DEA d'informatique. juin 1993. 63 p.

Blanchon H. (1991). *Problèmes de désambiguïsation interactive en TAO Personnelle.* Proc. L'environnement traductionnel ; La station de travail du traducteur de l'an 2001. Mons, Belgique. 25-27 avril 1991, vol. **1/1** : pp. 31-48.

Blanchon H. (1992). *A Solution to the Problem of Interactive Disambiguation.* Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. **4/4** : pp. 1233-1238.

Blanchon H. (1994a). *LIDIA-1 : une première maquette vers la TA interactive "pour tous".* Thèse. Université Joseph Fourier - Grenoble 1. 1994a. 321 p.

Blanchon H. (1994b). *Perspectives of DBMT for monolingual authors on the basis of LIDIA-1, an implemented mockup.* Proc. Coling-94. Kyoto, Japon. 5-9 août 1994, À paraître, 5p, vol. .

Boitet C. (1989). *Motivations, aims and architecture of the Lidia project.* Proc. MT SUMMIT II. Munich. 16-18 août 1989, vol. **1/1** : pp. 53-57.

Boitet C. (1990a). *Software and lingware engineering in recent (1980—90) classical MT : Ariane-G5 and BV/aero/F-E.* Proc. ROCLing-III (tutorials). Taipeh, Aug. 1990, vol. : pp. 21 p.

Boitet C. (1990b). *Towards Personnal MT : general design, dialogue structure, potential role of speech.* Proc. Coling-90. Helsinki. 20-25 Août 1990, vol. **3/3** : pp. 30-35.

Boitet C. & Blanchon H. (1993). *Dialogue-based MT for monolingual authors and the LIDIA project.* Proc. NLPRS'93. Fukuoka, Japon. 6-7 décembre 1993, vol. **1/1** : pp. 208-222.

Boitet C. & Blanchon H. (1994). *Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mock-up.* in Machine Translation. vol. **À paraître** : pp. 21.

Boitet C. & Seligman M. (1994). *The "Whiteboard" Architecture: a way to integrate heterogeneous components of NLP systems.* Proc. Coling-94. Kyoto, Japon. 5-9 août 1994, vol. **1/2** : pp. 426-430.

Brown R. D. (1989). *Augmentation.* in Machine Translation. vol. **4** : pp. 1299-147.

Brown R. D. & Nirenburg S. (1990). *Human-Computer Interaction for Semantic Disambiguation.* Proc. Coling-90. Helinki. 20-25 août 1990, vol. **3/3** : pp. 42-47.

Carbonell J. G. (1983). *The Role of User Modeling in Natural Language Interface Design.* Rap. Carnegie Mellon University. Rapport de recherche. n° CMU-CS-83-115. 29 mai 1983. 10 p.

Carré R., Dégremont J.-F., Gross M., Pierrel J.-M. & Sabah G. (1991). *Language humain et machine.* Presses du CNRS. Paris. ? p.

Chandler B., Holden N., Horsfall H., Pollard E. & McGee Wood M. (1987). *N-Tran Final Report. Alvey Project.* Rap. CCL/UMIST, Manchester. n° 87/9. 23 p.

Church K. & Patil R. (1982). *Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table.* in American Journal of Computional Lingusitics. vol. **8**(3-4) : pp. 139-149.

Frankish C., Jones D. & Hapeshi K. (1992). *Decline in accuracy of automatic speech recognition as a function of time on task: fatigue or voice drift?* in International Journal of Man-Machine Studies. vol. **36**(6) : pp. 797-816.

Fuchs C. (ed), (1985). *Aspects de l'ambiguïté et de la paraphrase dans les langues naturelles.* Peter Lang. Berne. 215 p.

Fuchs C. (1987). *L'ambiguïté et la paraphrase en linguistique.* in L'ambiguïté et la paraphrase. Centre de Publication de l'Université de Caen. Caen. pp. 15-20.

Goodman K. & Nirenburg S. (ed), (1991). *The KBMT Project: A case study in knowledge-based machine translation.* Morgan Kaufmann. San Mateo, California. 331 p.

Haverkort K. (1990). *Some reflections on ambiguity and its concequences for MT.* Proc. The Maastricht Session of the 1990 Maastricht-Lódz Duo Colloquium on "Translation and Meaning". Maastricht, The Netherlands. 4-6 janvier 1990, vol. : pp. 28-38.

Hirst G. (1987). *Semantic interpretation and the resolution of ambiguity.* Cambridge University Press. Cambridge. 263 p.

Huang X. (1990). *A Machine Translation System for the Target Language Inexpert.* Proc. Coling-90. Helsinki. 20-25 août 1990, vol. **3/3** : pp. 364-367.

Hutchins W. J. & Somers H. L. (1992). *An Introduction to Machine Translation.* Academic Press. 362 p.

Kay M. (1973). *The MIND system.* in Courant Computer Science Symposium 8: Natural Language Processing. Algorithmics Press, Inc. New York. pp. 155-188.

Lepage Y. (1985). Ambiguïtés et traduction automatique : les méthodes du GETA. in Brises. vol. **7** : pp. 62-71.

Lepage Y. (1988). *Ambiguities and Second Generation MT Systems.* Proc. First European Conference on Information Technology for Organisational systems. Athens. 16-20 mai 1988, vol. .

Maruyama H., Watanabe H. & Ogino S. (1990). *An Interactive Japanese Parser for Machine Translation.* Proc. Coling-90. Helsinki. 20-25 août 1990, vol. **2/3** : pp. 257-262.

Melby A. K. (1981). *Translators and Machines - Can they cooperate ?* in META. vol. **26**(1) : pp. 23-34.

Melby A. K. (1987). *On human-machine interaction in translation.* in Machine Translation. Cambridge University Press. Cambridge. pp. 145-154.

Melby A. K., Smith M. R. & Peterson J. (1980). *ITS : An Interactive Translation System.* Proc. Coling-80. Tokyo. 30 septembre-4 octobre 1980, vol. **1/1** : pp. 424-429.

Nijholt A. (1988). *Computers and Languages; Theory and Practice.* Elsevier Science Publishers B.V. Amsterdam. 482 p.

Papagaaij B. C. (1986). *Word Expert Semantics, an Interlingual Knowledge-Based Approach.* Floris Publications. Dordrecht, Holland. 240 p.

Sadler V. (1989). *Working with analogical semantics : Disambiguation techniques in DLT.* Floris Publications. Dordrecht, Holland. 256 p.

Saito H. (1992). *Interactive Speech Understanding.* Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. **3/4** : pp. 1053-1057.

Schubert K. (1986). *Linguistic and Extra-Linguistic Knowledge. A catalog of language-related rules and their computational application in machine translation.* in Computers and Translation. vol. **1**(3) : pp. 125-152.

Seligman M. & Boitet C. (1994). *A "whiteboard" architecture for automatic speech translation.* Proc. International Symposium on Spoken Dialogue. Waseda University, Tokyo. 1-12 novembre 1993, vol. : pp. 4.

Sérasset G. (1994). *Interlingual Lexical Organisation for Multilingual Lexical Databases in NADIA.* Proc. Coling-94. Kyoto, Japon. 5-9 août 1994, vol. **1/2** : pp. 278-282.

Tomita M. (1984). *Disambiguating Grammaticaly Ambiguous Sentences by Asking.* Proc. Coling-84. Stanford University, California. 2-6 juillet 1984, vol. **1/1** : pp. 476-480.

Tomita M. (1985). *Feasibility Study of Personal/Interactive Machine Translation System.* Proc. Conference on Theoretical and methodological Issues in Machine Translation of Natural Language. Colgate University, Hamilton, New York. 14-16 août 1985, vol. **1/1** : pp. 289-297.

Tomita M. (1986). *Sentence disambiguation by asking.* in Computers and Translation. vol. **1**(1) : pp. 39-51.

Tsutsumi T., Watanabe H., Maruyame H., Uramoto N., Morohashi M., Takeda K. & Nasukawa T. (1993). *Example-Based Approach to Machine Translation.* Proc. Premières journées franco-japonaise sur la traductionassistée par ordinateur. Ambassade de France au Japon, Tokyo, Japon. 15-16 mars 1993, vol. **1/1** : pp. 161-169.

Vauquois B. & Nédobejkine N. (1977). *Ambiguïtés de la langue écrite.* Rap. Action Thématique Programmée de Linguistique Générale, Table ronde organisée au CNRS. 11 janvier 1977. 8 p.

Wehrli É. (1990). *STS: An Experimental Sentence Translation System.* Proc. Coling-90. Helsinki. 20-25 août 1990, vol. **1/3** : pp. 76-78.

Wehrli É. (1991). *Pour une approche ineractive au problème de la traduction automatique.* Proc. L'environnement traductionnel ; La station de travail du traducteur de l'an 2001. Mons, Belgique. 25-27 avril 1991, vol. **1/1** : pp. 59-68.

Wehrli É. (1992). *The IPS System.* Proc. Coling-92. Nantes, France. 23-28 juillet 1992, vol. **3/4** : pp. 870-874.

Wehrli É. (1993). *Vers un système de traduction interactif.* in La traductique. Les presses de l'Université de Montréal, AUPELF/UREF. pp. 423-432.

Whitelock P. J., Wood M. M., Chandler B. J., Holden N. & Horsfall H. J. (1986). *Strategies for Interactive Machine translation : the experience and implications of the UMIST Japanese project.* Proc. Coling-86. Bonn. 25-29 août 1986, vol. **1/1** : pp. 25-29.

Wioland F. (1987). *Jointure et ambiguïté.* in L'ambiguïté et la paraphrase. Centre de Publication de l'Université de Caen. Caen. pp. 53-57.

Yamaguchi M., Takiguchi N., Kotani Y. & Nisimura H. (1993). *An Interactive Method for Semantic Disambiguation in Sentences by Selecting Examples.* Proc. NLPRS'93. Fukuoka, Japon. 6-7 décembre 1993, vol. : pp. 208-222.

Appendix

Appendix A

Ambiguity and natural language processing: a classification

In this chapter, we propose a classification of ambiguity which is a synthesis of several works we have studied.

Our more important information sources were: [Béjoint 1990 ; Carré, *et al.* 1991 ; Fuchs 1985 ; Fuchs 1987 ; Haverkort 1990 ; Hirst 1987 ; Hutchins & Somers 1992 ; Lepage 1985 ; Lepage 1988 ; Nijholt 1988 ; Sadler 1989 ; Schubert 1986 ; Vauquois & Nédobejkine 1977].

A.1. Oral-input-specific ambiguities

A.1.1. Homophony

Homophones are words with the same sound but different meanings.

Examples :	Remark :
Les grands mots entraînent toujours de grands maux. (A. France)	mots/maux

[ilpr̃aeso]	il prend le seau/le sceau/le sot
[ilparl]	ils parlent/il parle
[japresiɛ]	j'appréciai/j'appréciais

In French there are some phonological rules that normally distinguish the pronunciations of "j'appréciai" and "j'appréciais". In practice, we observe many variations relative to local usage or accents which eliminate the difference. This phenomenon occurs also in other languages.

A.1.2. Word segmentation

In this situation, the single segmèntation of an articulated string in words, phrases or sentences is not possible. For a more detailed account cf. [Wioland 1987]

Examples :	Remark :
[tynepøpalavwar]	Tu ne peux pas l'avoir/tu ne peux pas la voir.
[ilsɔ̃trɜzɑmi]	Ils sont treize amis./Ils sont très amis
[lamitoz]	l'amitose/la mitose
plant some more tulips	plant some / plan some some more / some ore tulips / two lips
We make all of our children.	We make all of our children / We may call of our children

A.2. Word ambiguity

A.2.1. Lexical ambiguity (word sense ambiguity)

A word can have several senses. The lexical ambiguity can be a property of a word considered within one language (internal lexical ambiguity) or can appear when comparing the words of one language with the words of another, for example while building a bilingual dictionary.

a. Internal lexical ambiguity

Examples :	Remark :
Il admire cette vedette.	vedette: star, bateau ou soldat
Est-ce que la pêche était bonne ?	pêche: fruit, pêcheur
L'aigle quitte son aire.	aire: nid, sa surface de chasse
C'était un vol extraordinairement risqué.	vol plané, vol à la tire
The box was in the pen.	pen: cage or writing instrument

b. Contrastive lexical ambiguity

Examples :	Remark :
"river" en anglais	"rivière" *ou* "fleuve" en français
"mur" en français	"Wand" *ou* "Mauer" en allemand ; "muro" *ou* "parete" en italien.

A.2.2. Categorial ambiguity (ambiguity of syntactic class)

This ambiguity appears when a word can be associated with several syntactic classes which refer generally to several lexical units.

Examples :	Remark :
Le pilote ferme la porte.	"le pilote porte" ou "le pilote ferme"
Devant cette somme, il hésite.	"somme due" *ou* "somme importante"
Un savant compromis a été arrêté hier.	"savant arrêté" *ou* "compromis trouvé"
They are cooking apple.	What are they cooking in the kitchen? What kind of apple are thoses?
Flying planes can be dangerous.	Dangerous to the person flying or to the people on the ground

A.2.3. Morphological ambiguity

The morphology of the word do not allow to choose the lexical features to be associated with the word.

Examples :	Remark :
L'artiste lit le livre.	un artiste *ou* une artiste
Je finis subitement.	indicatif : présent *ou* passé simple
I read.	présent *ou* passé
He saw the sheep.	"one sheep" *ou* "several sheep"

A.3. Structural ambiguity

From a set of words, the phrases can be constructed and interrelated differently according to syntagmatic properties.

A.3.1. Compound nouns, idioms

Examples :	Remark :
Il convient avec le gardien de l'ordre du jour.	"gardien de l'ordre" *ou* "ordre du jour"
Vous sacrifiez l'artiste à la mode que vous pensez indigne.	"artiste à la mode" *ou* "mode indigne"

a thin can opener	(thin can) opener *ou* thin (can opener)

A.3.2. Structure of the grammatical phrases

This kind of ambiguity is almost the same as the ambiguity of compound nouns. The whole sentence is not misunderstood, but only phrases (particularly noun phrases). It can be divided into two categories depending on whether coordination or subordination is implied.

a. Coordination

Examples :	Remark :
On étudie la structure du réseau et des investissements.	"structure du réseau et structure des investissements" *ou* "investissements et structure du réseau"
Il prends des crayons et des classeurs noirs.	"crayons noirs et classeurs noirs" *ou* "classeurs noirs et crayons"
Les neutrons et les protons rapides...	"protons rapides" ou "neutrons rapides et protons rapides"

b. Subordination

Examples :	Remark :
Jean rend ce livre illisible.	Le livre était-il illisible avant ?
Le magistrat juge les enfants coupables.	"enfants coupables jugés" *ou* "enfants jugés coupables"
Le capitaine a rapporté un vase de Chine.	"vase chinois" *ou* "vase quelconque rapporté de Chine"
Le colonel X a été trouvé pendu par sa femme. (Le Monde, 1964)	"sa femme a pendu le capitaine" *ou* "sa femme a trouvé le capitaine pendu"
L'école de cuisine lyonnaise est ouverte.	"école lyonnaise" *ou* "cuisine lyonnaise"
Elle épouse un professeur de droit anglais.	"professeur anglais" *ou* "droit anglais"

A.4. Logic and/or semantic relation ambiguity (functional ambiguity)

The structural ambiguities relative to the logic constituents come, most of the time, from unsolved morphological ambiguities. They are especially frequent in languages where the word order is not fixed for each logical term. This kind of ambiguity is called functionnal ambiguity by Sadler [Sadler 1989].

Examples :	Remark :
Pierre fait manger les poulets.	les poulets mangent ou sont mangés ?
Jean a acheté un livre à Pierre.	Pierre est vendeur *ou* un livre pour Pierre
Pierre fait porter des chocolats à Lucie.	Lucie porte ou reçoit les chocolats
Je vote pour ma mère.	en faveur ou à la place de ma mère

Je vous parle de la Tour Eiffel.	depuis ou à propos de la Tour Eiffel
Quel auteur cite ce conférencier ?	un auteur cite ou un conférencier cite
J'ignore quels ennemis redoutaient les soldats.	"les soldats redoutent des ennemis "ou "les ennemis redoutent les soldats"

A.5. Anaphora

The word anaphora stands here for pronoun reformulation and ellipsis.

A.5.1. Pronominal or adjectival repetition

Which element has been elided, pronominalized?

Examples :	Remark :
Les médecins ont soigné les malades, ils sont très contents.	"les malades sont contents" *ou* "les médecins sont contents"
La fermière vend sa vache parce qu'elle est malade	"la vache est malade" *ou* "la fermière est malade"
The soldiers fired at the women and they fell.	The soldier fell or the women?

A.5.2. Ellipsis

There is an ellipsis when a part of a sentence or a phrase is elided. The ellipsis can be simple, if only one word or group of words has been elided, or complex, if several groups have been elided.

a. Simple

Examples :	Remark :
Elle aime son fils plus que son mari.	"Elle aime plus son fils qu'elle n'aime son mari." *ou* "Son mari aime moins son fils qu'elle n'aime ce fils."
Il traite son prochain comme un ennemi.	"... comme si c'était un ennemi." *ou* "... comme un ennemi traite son prochain."
Lorsqu'une belle lune met fin à une morne journée, la suivante sera plus belle.	"la lune suivante sera plus belle" *ou* "la journée suivante sera plus belle"

b. On a compound word

This form of ellipsis is a particular case of polysemia. Indeed, when some part of a compound noun has been elided, to alleviate or shorten the text, the word is often replaced by its head. The head then becomes a polysemic occurrence.

Examples

Entretien de la centrale inertielle. Démonter puis déposer la centrale. Vérifier le guide d'amorce de l'échangeur. Si nécessaire, changer le guide et remonter la centrale.

Un micro-ordinateur portable → un portable

Une phrase transitive → une transitive

Un roman de la série noire → un série noire

A.6. Other ambiguities

A.6.1. Definiteness, generality

Examples :	Remark :
Elle veut épouser un milliardaire.	"Jean qui est milliardaire" *ou* "n'importe qui pourvu qu'il soit milliardaire"
Jean doit aller à l'hôpital.	"cet hôpital précis" *ou* "un hôpital quelconque"

A.6.2. Quantifier scope

Examples :	Remark :
Un fleuve traverse chaque pays européen.	"des fleuves différents traversent chaque pays, un par pays" *ou* "un même fleuve traverse tous les pays".

Those two last kinds of ambiguity can often be preserved by the translation. But there are some cases in which they should be solved.

Examples :

determination for Russian → French or Japanese → French,

aspect for French → Japanese.

A.6.3. Modality

Examples :	Remark :
Il doit venir.	He should, might, have to come.
He should come soon.	Kare ga sugu kitara ii to omou. Kare ga sugu kuru hazu da.

A.6.4. Intention, IFT, speech act

Examples :	Remark :
Hai.	Yes, that's right. Okay, I will. Yes? What do you want?

Appendix B

The structure produced by the analyzer

The analyzer produces a structure called mmc for Multisolution, Multilevel and Concrete.

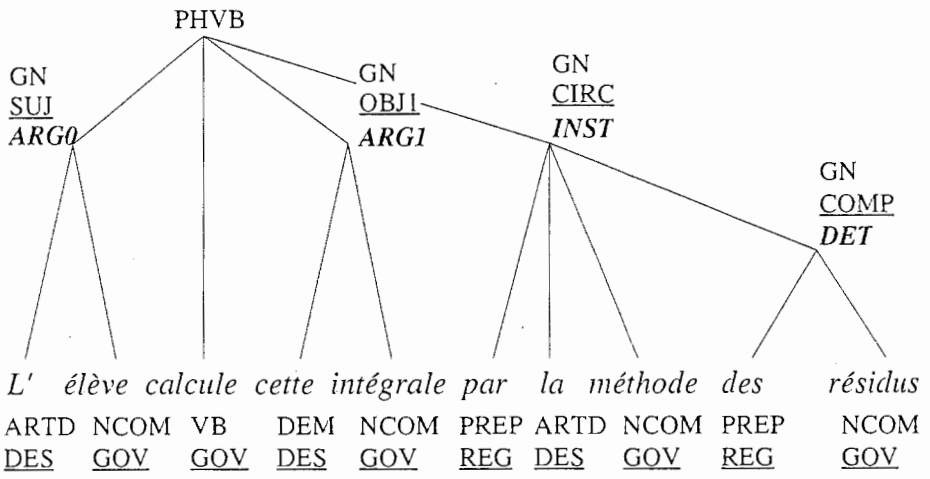


Figure B.1: an example of an mc-structure (multilevel, concrete)

Multisolution means that the analyzer produces every analysis fitting with the syntagmatic, syntactic and logico-semantic model of the grammar.

Multilevel means that the same structure consists of three levels of linguistic interpretation, namely the level of syntactic and syntagmatic classes, the level of syntactic functions and the level of logic and semantic relations.

Finally, the structure is said to be concrete because the original utterances can be reconstructed by a simple left-to-right reading of each analysis tree (mc-structure) occurring in the mmc-structure.

The previous figure is the multilevel and concrete structure produced for the sentence "L'élève calcule cette intégrale par le méthode des résidus[86]."

[86] The student calculate this integral by the method of the residues.

Appendix C

Labels used in the mmc-structure

This appendix lists a subset of the set of labels used in the mmc-structure produced by the analyzer in the framework of the LIDIA-1.0 mockup. The tree structures given in this document use these labels for the most part. The other labels used by the analyzer are currently irrelevant for the disambiguation process.

C.1. First level of labeling

C.1.1. Syntactic classes

N	NOM	NOUN
V	VERBE	VERB
ADJ	ADJECTIF	ADJECTIVE
ADV	ADVERBE	ADVERB
CRD	CARDINAL	CARDINAL
D	DÉICTEUR	DEICTIC
R	REPRÉSENTANT	PRONOUN
S	SUBORDONNANT	SUBORDINATING PARTICLE
C	COORDONNANT	COORDINATION PARTICLE
P	PONCTUATION	PUNCTUATION
PREF	PRÉFIXE	PREFIX
INC	MOT INCONNU	UNKNOWN WORD
NA	TEXTE NON ALPHABÉTIQUE	NON ALPHABETIC TEXT (EX: EQUATION)

C12 Syntagmatic classes

GN	GROUPE NOMINAL	NOUN PHRASE
GADJ	GROUPE ADJECTIVAL	ADJECTIVAL PHRASE
GCARD	GROUPE CARDINAL	CARDINAL PHRASE
GORD	GROUPE ORDINAL	ORDINAL PHRASE
GADV	GROUPE ADVERBIAL	ADVERBIAL PHRASE
NV	NOYAU VERBAL	VERBAL PHRASE (KERNEL)
PHVB	PROPOSITION VERBALE	VERBAL CLAUSE
PHINF	PROPOSITION INFINITIVE	INFINITIVE CLAUSE
PHPART	PROPOSITION PARTICIPIALE	PARICIPLE CLAUSE
PHREL	PROPOSITION RELATIVE	RELATIVE CLAUSE
PHSUB	PROPOSITION SUBORDONNÉE	SUBORDINATED CLAUSE

C2 Second level of labeling: syntactic functions

GOV	GOUVERNEUR DU GROUPE	HEAD
AUX	AUXILIAIRE	AUXILIARY
SUJ	SUJET	SUBJECT
OBJ	OBJET (ARG1 OU ARG2)	OBJECT
CPAG	COMPLÉMENT D'AGENT	AGENT COMPLEMENT
CIRCP	CIRCONSTANT PRIVILÉGIÉ	PRIVILEGED CIRCUMSTANT
CIRC	CIRCONSTANT	CIRCUMSTANT
COMP	COMPLÉMENT DE NOM	NOUN COMPLEMENT
ATSUJ	ATTRIBUT DU SUJET	SUBJECT COMPLEMENT
ATOBJ	ATTRIBUT DE L'OBJET	OBJECT COMPLEMENT
ATG	ATTRIBUT DU GOUVERNEUR	HEAD COMPLEMENT
ATGR	ATTRIBUT DU GROUPE	PHRASE COMPLEMENT
COORD	COORDINATION	COORDINATION
ENUM	ÉNUMÉRATION	ENUMERATION
NUMERO	NUMÉROTATION DES PARAGRAPHES	PARAGRAPH NUMBER
DES	DÉSIGNATEUR DE GROUPE	DETERMINER
REG	RÉGISSEUR DE GROUPE	SUBORDINATING PARTICLE
JUXT	JUXTAPPOSITION	JUXTAPOSITION
RFDG	RÉFÉRENCE AU DEGRÉ	REFERENCE TO DEGREE
LXAX	LÉXÈME AUXILIAIRE	AUXILIARY WORD
APP	APPOSITION	APPOSITION
VOCATIF		VOCATIVE
INCL	INCLUSION	INCLUSION

C3 Third level of labeling

C31 Logic relations (LR)

ARG0	ARGUMENT 0	ARGUMENT 0
ARG1	ARGUMENT 1	ARGUMENT 1
ARG2	ARGUMENT 2	ARGUMENT 2
ARG01	ARGUMENT 0 ET 1	ARGUMENT 0 AND 1 (SYMMETRICAL VERB)
ARG02	ARGUMENT 0 ET 2	ARGUMENT 0 AND 2 (SYMMETRICAL VERB)
ARG12	ARGUMENT 1 ET 2	ARGUMENT 1 AND 2 (SYMMETRICAL VERB)
TRL1	ATTRIBUT DE L'ARGUMENT 1	COMPLEMENT OF THE ARG1
GRA1	RELATION INVERSE DE ARG1	OPPOSITE RELATION OF THE ARG1
GRA2	RELATION INVERSE DE ARG2	OPPOSITE RELATION OF THE ARG2
GRA0	RELATION INVERSE DE ARG0	OPPOSITE RELATION OF THE ARG0
TRL0	ATTRIBUT DE L'ARG0	COMPLEMENT OF THE ARG0
ID	RL IDENTIQUE À CELLE DU PÈRE	SAME LR AS THE MOTHER

C32 Semantic relations (SR)

ALTERN	ALTERNATIVE	entre manger et boire	ALTERNATIVE
METHODE	MÉTHODE	par itération	METHOD
CAUSE	CAUSE		CAUSE
MATIERE	MATIÈRE CONSTITUANTE	en caoutchouc	MATTER
GENITIF	APPARTENANCE	le siège du pilote: son siège	POSSESSION
COND	CONDITION		CONDITION
FINAL	FINALITÉ D'UN OBJET = CE POURQUOI IL EST FAIT	nécessaire pour obtenir table de travail	FINALITY
BENEF	BÉNÉFICIAIRE		BENEFICIARY
CONSEQ	CONSÉQUENCE		CONSEQUENCE
THEME	CE DONT ON PARLE, RELATIF À	un livre sur la politique	THEME
INST	INSTRUMENT	avec une pince	INSTRUMENT
CTXT	CONTEXTE DANS LEQUEL SE DÉROULE OU DOIT SE DÉROULER UNE ACTION	faire qque chose avion sur roues	CONTEXT
MANIERE	MANIÈRE	visser, en prenant garde à ...	MANNER
ACCOMP	ACCOMPAGNEMENT ET NON ACCOMPAGNEMENT	avec le pilote, sans le pilote	ACCOMPANIMENT AND NON-ACCOMPANIMENT

ANALOG	ANALOGIE	il travaille comme un fou	ANALOGY
CONCESS	CONCESSION	bien que ...	CONCESSION
QFIER	QUANTIFIEUR	quelques (pommes) (un sac) de 3 kg	QUANTIFIER
OBJQTF	OBJET QUANTIFIÉ = RELATION INVERSE DE QFIER	3 kg de sucre	QUANTIFIED OBJECT
MODUL	MODULATION D'UNE QUANTIFICATION	environ 3 heures plus de 5 sacs	MODULATION OF A QUANTIFICATION
UNDE	LIEU D'OÙ L'ON VIENT ou D'OÙ L'ON PREND QQ CHOSE ou BORNE PASSÉ DANS LE TEMPS (INTERVALLE)	il vient de paris depuis 3 jours après mon départ	PLACE WHERE WE COME FROM or FROM WHERE WE TAKE STG or LIMIT IN THE PAST (INTERVAL)
UBI	LIEU OÙ L'ON EST ou LIEU OÙ SE PASSE L'ACTION ou DATE PONCTUELLE	marcher dans le hangar hier, il y a 3 jours	PLACE WHERE WE ARE or PLACE OF THE ACTION or PUNCTUAL DATE
QUO	LIEU OÙ L'ON VA ou OÙ L'ON MET QQUE CHOSE ou BORNE FUTURE DANS LE TEMPS (INTERVALLE)	introduire dans --- avant lundi jusqu'au 10 juin	PLACE WHERE WE GO or PLACE WHERE WE PUT STG or LIMIT IN THE FUTURE
QUA	LIEU PAR OÙ L'ON PASSE ou SIMULTANÉITÉ PAR RAPPORT À L'ACTION (TEMPS)	passer par la fene^tre pendant le remplissage	PLACE WHERE WE ARE or SIMULTANEITY IN RELATION TO THE ACTION
PART	PARTIES CONSTITUANTES D'UN TOUT	un tas de feuilles mortes	PARTS OF A WHOLE
TRAP	RELATION INVERSE DE PART	le bouchon du réservoir	OPPOSIT OF PART
QUM	QUANTUM, QUANTIFICATION PAR UN GROUPE NOMINAL NON INTRODUIT PAR UNE PRÉPOSITION	(il pèse) 10 kg (il a acheté cette voiture) 50000 francs	QUANTUM, QUANTIFICATION BY A NOUN PHRASE
SAUV	RS BIDON, UTILE EN PARTICULIER POUR LXAX, DES --		NOT USED
QUALF	QUALIFICATION FORTE (GN SANS DÉTERMINATION)	le fromage de chèvre, la table à roulettes	GENERIC QUALIFICATION
QUAL	TOUS LES AUTRES COMPLÉMENTS		ALL THE OTHER COMPLEMENTS
ID			SAME SR AS THE MOTHER