TR-IT-0112

# Towards ambiguity labelling for the study of interactive disambiguation methods

Christian BOITET & Mutsuko TOMOKIYO

1995.4.27

This report has been prepared in the context of the MIDDIM project (ATR-CNRS). It introduces the concept of "ambiguity labelling", and proposes a precise text processor oriented format for labelling "pieces" such as dialogues and texts. Several notions concerning ambiguities are made precise, and many examples are given. The ambiguities labelled are meant to be those which state-of-the-art speech analyzers are believed not to be able to solve, and which would have to be solved interactively to produce the correct analysis. The proposed labelling has been specified with a view to store the labelled pieces in a data base, in order to estimate the frequency of various types of ambiguities, the importance to solve them in the envisaged contexts, the scope of disambiguation decisions, and the knowledge needed for disambiguation. A complete example is given. Finally, an equivalent data base oriented format is sketched.

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

# Table of Contents



-o-o-o-o-o-o-o-o-o-o-o-

# Towards ambiguity labelling for the study of interactive disambiguation methods

Christian BOITET & Mutsuko TOMOKIYO

visiting researcher from
GETA, CLIPS, IMAG (UJF, CNRS & INPG)
BP 53, F-38041 Grenoble Cedex 9

ATR Interpreting Telecommunications
2-2 Hikari-dai, Seika-cho, Kyoto 619-02

## Abstract

This report has been prepared in the context of the MIDDIM project (ATR-CNRS). It introduces the concept of "ambiguity labelling", and proposes a precise text processor oriented format for labelling "pieces" such as dialogues and texts. Several notions concerning ambiguities are made precise, and many examples are given. The ambiguities labelled are meant to be those which state-of-the-art speech analyzers are believed not to be able to solve, and which would have to be solved interactively to produce the correct analysis. The proposed labelling has been specified with a view to store the labelled pieces in a data base, in order to estimate the frequency of various types of ambiguities, the importance to solve them in the envisaged contexts, the scope of disambiguation decisions, and the knowledge needed for disambiguation. A complete example is given. Finally, an equivalent data base oriented format is sketched.

## Introduction

As has been argued in detail in [4, 6, 10], interactive disambiguation technology must be developed in the context of research towards practical Interpreting Telecommunications systems as well as high-quality multitarget text translation systems. In the case of speech translation, this is because the state of the art in the foreseeable future is such that a black box approach to spoken language analysis (speech recognition plus linguistic parsing) is likely to give a correct output for no more than 50 to 60% of the utterances ("Viterbi consitency" [8])[1], while users would presumably require an overall success rate of at least 90% to be able to use such systems at all. However, the same spoken language analyzers may be able to produce sets of outputs containing the correct one in about 90% of the cases ("structural consistency" [8])[2]. In the remaining cases, the system would be unable to analyze the input, or no output would be correct.

Interactive disambiguation by the users of the interpretation or translation systems is then seen as a practical way to reach the necessary success rate.

*It must be stressed that interactive disambiguation is not to be used to solve all ambiguities. On the contrary, as many ambiguities as possible should be reduced automatically. The remaining ones should be solved by interaction as far as practically possible. What is left would have to be reduced automatically again, by using preferences and defaults.*

In other words, this research is complementary to the research in automatic disambiguation. Our stand is simply that, given the best automatic methods currently available, which use syntactic and semantic restrictions, limitations of lexicon and word senses by the generic task at hand, as well as prosodic and pragmatic cues, too many ambiguities will remain after automatic analysis, and the "best" result will not be the correct one in too many cases.

---

[1] According to a study by Cohen & Oviatt, the combined success rate is bigger than the product of the individual success rates by about 10% in the middle range. Using a formula such as S2 = S1*S1 + (1-S1)*A with A=20%, we get:

| SR of 1 component (S1) | 40% | 45% | 50% | 55% | 60% | 65% | 70% | 75% | 80% | 85% | 90% | 95% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SR of combination (S2) | 28% | 31% | 35% | 39% | 44% | 49% | 55% | 61% | 68% | 75% | 83% | 91% | 100% |

50~60% overall Viterbi constitency corresponds then to 65~75% individual success rate, which is already optimistic.

[2] According to the preceding table, this corresponds to a structural consistency of 95% for each component, which seems impossible to attain by strictly automatic means in practical applications involving general users.

We suppose that the system will use a state-of-the-art language-based speech recognizer and multilevel analyzer, producing syntactic, semantic and pragmatic information. We leave open two possibilities:

- an expert system specialized in the task at hand may be available.
- an expert human interpreter/translator may be called for help over the network.

The questions we want to address in this context are the following:

- *what kinds of ambiguities (unsolvable by state-of-the-art speech analyzers) are there in dialogues and texts to be handled by the envisaged systems?*
- *what are the possible methods of interactive disambiguation, for each ambiguity type?*
- *how can a system determine whether it is important or not for the overall communication goal to disambiguate a given ambiguity?*
- *what kind of knowledge is necessary to solve a given ambiguity, or, in other word, whom should the system ask: the user, the interpreter, or the expert system, if any?*
- *in a given dialogue or document, how far do solutions to ambiguities carry over: to the end of the piece, to a limited distance, or not at all?*

In order to answer these questions, it seems necessary to build a data base of ambiguities occurring in the intended contexts. In this report, we are not interested in any specific data base management software, but in the collection of data, that is, in "ambiguity labelling".

First, we make more precise several notions, such as ambiguous representation, ambiguity, ambiguity kernel, ambiguity type, etc. Second, we specify the attributes and values used for manual labelling, and give a text processor oriented format. Third, we give a complete example of ambiguity labelling of a short dialogue, with comments. Finally, we define a data-base oriented exchange format.

# I. A formal view of ambiguities

## I.1 Levels and contexts of ambiguities

### 1.1 Three levels of granularity for ambiguity labelling

First, we distinguish three levels of granularity for considering ambiguities.

There is an ambiguity at the level of a dialogue (resp. a text) if it can be segmented in at least two different ways into turns (resp. paragraphs). We speak of *ambiguity of segmentation into turns* or *into paragraphs.*

There is an ambiguity at the level of a turn (resp. a paragraph) if it can be segmented in at least two different ways into utterances[1]. We speak of *ambiguity of segmentation into utterances.*

There is an ambiguity at the level of an utterance if it can be analyzed in at least two different ways, whereby the analysis is performed in view of translation into one or several languages in the context a a certain generic task. There are various types of *utterance-level ambiguities.*

Ambiguities of segmentation into paragraphs may occur in written texts, if, for example, there is a separation by a <new_line> character only, without <line_feed> or <paragraph>. They are much more frequent and problematic in dialogues.

For example, in ATR's transcriptions of Wizard of Oz interpretations dialogues [22], there are an agent (A), a client (C), and an interpreter (I). In many cases, there are two successive turns of I, one in Japanese and one in English. Sometimes, there are even 3 in a row (p. 32: J-E-J, p. 33: E-J-J). If I does not help the system by pressing a button, this ambiguity will force the system to do language identification every time there may be a change of language. There are also cases of two successive turns by C (p. 27, E), and even 3 by A (p. 52, J) and I (p. 55, J-E-J, p. 80, E-E-J) or 4 (I, p. 99, E-J-E-

[1] We use the term "utterance" for dialogues and texts, to stress that the "units of analysis" are not always sentences, but may be titles, interjections, etc.

J). Studying these ambiguities is important for discourse analysis, which assumes a correct analysis in terms of turns. Also, if successive turns in the same language are collapsed, this may add ambiguities of segmentation into utterances, leading in turn to more utterance-level ambiguities.

Ambiguities of segmentation into utterances are very frequent, and most annoying, as we assume that the analyzers will work utterance by utterance, even if they have access to the result of processing of the preceding context. There are for instance several examples of "right !? now !? turn left...". Or ([22], p. 50): "OK !? so go back and is this number three !? right there !? shall I wait here for the bus?".

An utterance may be spoken or written, may be a sentence, a phrase, a sequence of words, syllables, etc. In the usual sense, there is an ambiguity in an utterance if there are at least two ways of *understanding* it. This, however, does not give us a precise criterion for defining ambiguities, and even less so for labelling them and storing them as objects in a data base. Because human understanding heavily depends on the context and the communicative situation, it is indeed a very common experience that something is ambiguous for one person and not for another.

Hence, we prefer to say that *an utterance is ambiguous if it has an ambiguous representation in some formal representation system.* We return to that later.

### 1.2 Task-derived limitations on utterance-level ambiguities

As far as utterance-level ambiguities are concerned, we will consider only those which we feel should be produced by any state-of-the-art analyzer constrained by the task. For instance, we should not consider that "good morning" is ambiguous" with "good mourning", in a conference registration task. It could be different in the case of funeral arrangements.

Because the analyzer is supposed to be state-of-the-art, "help" should not give rise to the possible meaning "help oneself" in "can I help you". Knowledge of the valencies and semantic restrictions on arguments of the verb "help" should eliminate this possibility.

In the same way, "Please state your phone number" should not be deemed ambiguous, as no complete analysis should allow "state" to be a noun, or "phone" to be a verb. That could be different in a context where "state" could be construed as a proper noun, "State", for example in a dialogue wher the State Department is involved.

However, we should consider as ambiguous such cases as: "Please state (N/V) office phone number" (p. 33), where "phone" as a verb could be eliminated on grammatical grounds, but not "state office phone" as a noun, with "number" as a verb in the imperative form. The case would of course be different if the transcription would contain prosodic marks, but the point would continue to hold in general.

### 1.3 Necessity to consider utterance-level ambiguities in the context of full utterances

Let us take another example. Consider the utterance:

(1) Do you know where the international telephone services are located?

The underlined fragment has an ambiguity of attachment, because it has two different "skeleton" [8] representations:

[international telephone] services / international [telephone services]

As a title, this sequence presents the same ambiguity. However, it is not enough to consider it in isolation. Take for example:

(2) The international telephone services many countries.

The ambiguity has disappeared! It is indeed frequent that an ambiguity relative to a fragment appears, disappears and reappears as one broadens its context in an utterance. For example, in

(3) The international telephone services many countries have established are very reliable.

the ambiguity has reappeared. From the examples above, we see that, in order to define properly what an ambiguity is, we must consider the fragment *within an utterance,* and clarify the idea that the fragment is the smallest (within the utterance) where the ambiguity can be observed.

## I.2 Representation systems

### 2.1 Types of formal representation systems

Classical representation systems are based on lists of binary features, flat or complex attribute structures (property lists), labeled or decorated trees, various types of feature-structures, graphs or networks, and logical formulae.

What is an "ambiguous representation"? This question is not as trivial as it seems, because it is often not clear what we exactly mean by "the" representation of an utterance. In the case of a classical context-free grammar G, shall we say that *a* representation of U is *any* tree T associated to U via G, or that it is the *set* of all such trees? Usually, linguists say that U has *several* representations with reference to G.

But if we use f-structures with disjunctions, U will always have one (or zero!) associated structure S. Then, we would like to say that S is ambiguous if it contains at least one disjunction. Returning to G, we might then say that "the" representation of U is a disjunction of trees T.

In practice, however, developers prefer to use hybrid data structures to represent utterances. Trees decorated with various types of structures are very popular. For speech and language processing, lattices bearing such trees are also used, which means at least 3 levels at which a representation may be ambiguous.

### 2.2 Computable representations and "reasonable" analyzers

Now, we are still left with two questions:

1) which representation system(s) do we choose?
2) how do we determine the representation or representations of a particular utterance in a specific representation system?

The answer to the first question is a practical one. The representation system(s) must be fine-grained enough to allow the intended operations. For instance, text-to-speech requires less detail than translation. On the other hand, it is counter-productive to make too many distinctions. For example, what is the use of defining a system of 1000 semantic features if no system and no lexicographers may assign them to terms in an efficient and reliable way? There is also a matter of taste and consensus. Although different representation systems may be formally equivalent, researchers and developers have their preferences. Finally, we should prefer representations amenable to efficient computer processing.

As far as the second question is concerned, two aspects should be distinguished. First, the consensus on a representation system goes with a consensus on its semantics. This means that people using a particular representation system should develop guidelines enabling them to decide which representations an utterance should have, at each level, and to create them by hand if challenged to do so. Second, these guidelines should be refined to the point where they may be used to specify and implement a parser producing all and only the intended representations for any utterance in the intended domain of discourse.

A "computable" representation system is a representation system for which a "reasonable" parser can be developed.

A "reasonable" parser is a parser such as:

- its size and time complexity are tractable over the class of intended utterances;
- if it is not yet completed, assumptions about its ultimate capabilities, especially about its disambiguation capabilities, are realistic given the state of the art.

Suppose, then, that we have defined a computable representation. We may not have the resources to build an adequate parser for it, or the one we have built may not yet be adequate. *In that case, given the fact that we are specifying what the parser should and could produce, we may anticipate and say that an utterance presents an ambiguity of such and such types.* This only means that we expect that an adequate parser will produce an ambiguous representation for the utterance at the considered level.

### 2.3 Expectations for a system of manual labelling

Our manual labelling should be such that:

- it is compatible with the representation systems used by the actual or intended analyzers.
- it is clear and simple enough for linguists to do the labelling in a reliable way and in a reasonable amount of time.

Representation systems may concern one or several levels of linguistic analysis. We will hence say that an utterance is phonetically ambiguous if it has an ambiguous phonetic representation, or if the phonetic part of its description in a "multilevel" representation system is ambiguous, and so forth for all the levels of linguistic analysis, from phonetic to orthographic, morphological, morphosyntactic, syntagmatic, functional, logical, semantic, and pragmatic.

In the labelling, we should only be concerned with the final result of analysis, not in any intemediate stage, because we want to retain only ambiguities which would remain unsolved after the complete automatic analysis process has been performed.

## I.3 Ambiguous representations

A representation will be said to be ambiguous if it is *multiple* or *underspecified.*

### 3.1 Proper representations

In all known representation systems, it is possible to define "proper representations", extracted from the usual representations, and ambiguity-free.

For example, if we represent "We read books" by the unique decorated dependency tree:

```
[["We" ((lex "I-Pro") (cat pronoun) (person 1) (number plur)…)]
"read" ((lex "read-V") (cat verb) (person 1) (number plur) (tense {pres past})…)
 ["books" ((lex "book-N") (cat noun)…)]]
```

there would be 2 proper representations, one with `(tense pres)`, and the other with `(tense past)`.

For defining the proper representations of a representation system, it is necessary to specify which disjunctions are exclusive, and which are inclusive.

*Proper and multiple representations*

A representation in a formal representation system is *proper* if it contains no exclusive disjunction.

The *set of proper representations associated to a representation R,* is obtained by expanding all exclusive disjunctions of R (and eliminating duplicates). It is denoted here by *Proper(R).*

R is *multiple* if |Proper(R)|>1. R is multiple if (and only if) it is not proper.

### 3.2 Underspecified representations

A proper representation P is *underspecified* if it is undefined with respect to some necessary information.

There are two cases: the information may be specified, but its value is unknown, or it is missing altogether.

The first case often happens in the case of anaphoras: `(ref ?)`, or for information which has not been computed, e.g. `(task_domain ?)`, `(decade_of_month ?)`, but which is necessary for translating in at least one of the considered target languages.

It is quite natural to consider this as ambiguous. For example, an anaphoric reference should be said to be ambiguous

- if several possible referents appear in the representation, which will give rise to several proper representations,
- and also if the referent is simply marked as unknown, which causes no disjunction.

The second case may never occur in representations such as Ariane-G5 decorated trees, where all attributes are always present in each decoration. But, in a standard f-structure, there is no way to force the presence of an attribute, so that a necessary attribute may be missing: then, `(ref ?)` is equivalent to the absence of the attribute `ref`.

For any formal representation system, then, we must specify what the "necessary information" is. Contrary to what is needed for defining Proper(R), this may vary with the intended application.

### 3.3 Ambiguous representations

Our final definition is now simple to state.

A representation R is *ambiguous* if it is multiple or if Proper(R) contains an underspecified P.

## I.4 Scope, occurrence, kernel and type of ambiguity

### 4.1 Informal presentation

Although we have said that ambiguities have to be considered in the context of the utterances, it is clear that a sequence like "international telephone services" is ambiguous in the same way in utterances (1) and (3) above. We will call this an "ambiguity kernel", and reserve the term of "ambiguities" for what we will label, that is, occurrences of ambiguities. The distinction is the same as that between dictionary words and text words.

It also clear that another sequence, such as "important business addresses", would present the same sort of ambiguity in analogous contexts. This we want to define as "ambiguity type". In this case, linguists speak of "ambiguity of attachment", or "structural ambiguity". Other types concern the acceptions (word senses), the functions (syntactic or semantic), etc. Our list will be given with the specification of the labelling conventions.

Ambiguity patterns are more specific kinds of ambiguity types, usable to trigger disambiguation actions, such as the production of a certain kind of disambiguating dialogue. For example, there may be various patterns of structural ambiguities.

### 4.2 Scope of an ambiguity

We take it for granted that, for each considered representation system, we know how to define, for each fragment V of an utterance U having a proper representation P, the part of P which represents V.

For example, given a context-free grammar and an associated tree structure P for U, the part of P representing a substring V of U is the smallest sub-tree Q containing all leaves corresponding to V. Q is not necessarily the whole subtree of P rooted at the root of Q.

Conversely, for each part Q of P, we suppose that we know how to define the fragment V of U represented by Q.

#### a. Scope of an ambiguity of underspecification

Let P be a proper representation of U. Q is a *minimal underspecified part* of P if it does not contain any strictly smaller underspecified part Q'.

Let P be a proper representation of U and Q be a minimal underspecified part of P. The *scope* of the ambiguity of underspecification exhibited by Q is the fragment V represented by Q.

In the case of an anaphoric element, Q will presumably correspond to one word or term V. In the case of an indeterminacy of semantic relation (deep case), e.g. on some argument of a predicate, Q would correspond to a whole phrase V.

b. Scope of an ambiguity of multiplicity

A fragment V presents an ambiguity of multiplicity n (n≥2) in an utterance U if it has n different proper representations which are part of n or more proper representations of U.

V is an *ambiguity scope* if it is minimal relative to that ambiguity. This means that any strictly smaller fragment W of U will have strictly less than n associated sub-representations (at least two of the representations of V are be equal with respect to W).

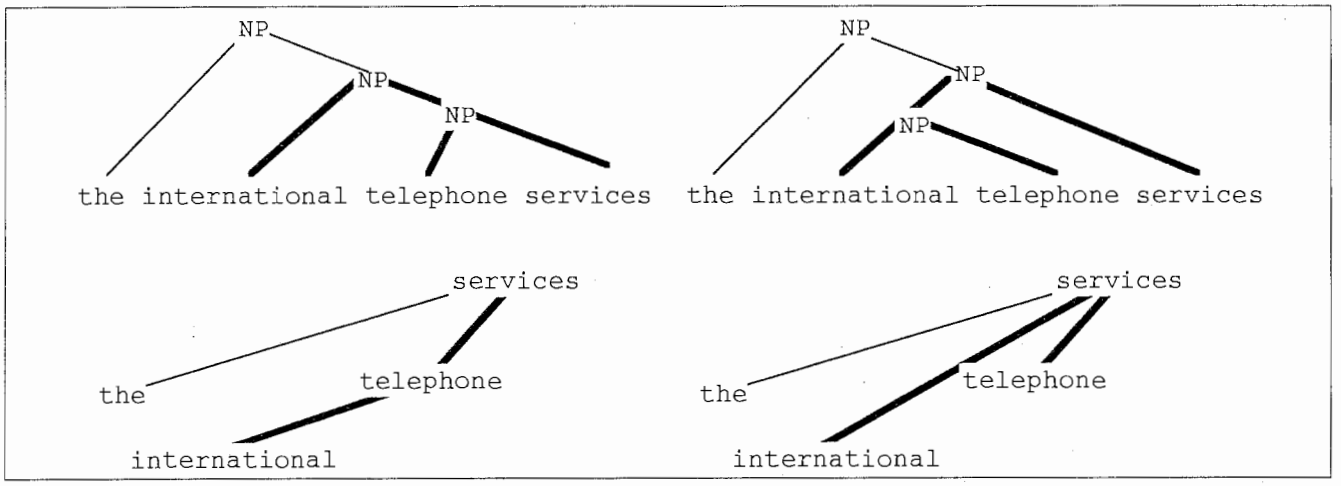
In example (1) above, then, the fragment "the international telephone services", together with the two skeleton representations

```
the [international telephone] services / the international [telephone services]
```

is not minimal, because it and its two representations can be reduced to the subfragment "international telephone services" and its two representations (which are minimal).

This leads us to consider that, in syntactic trees, the representation of a fragment is not necessarily a "horizontally complete" subtree.

In the case above, for example, we might have the configurations given in the figure below. In the first pair (constituent structures), "international telephone services" is represented by a complete subtree. In the second pair (dependency structures), the representing subtrees are not complete subtrees of the whole tree.




## 4.3 Occurrence and kernel of an ambiguity

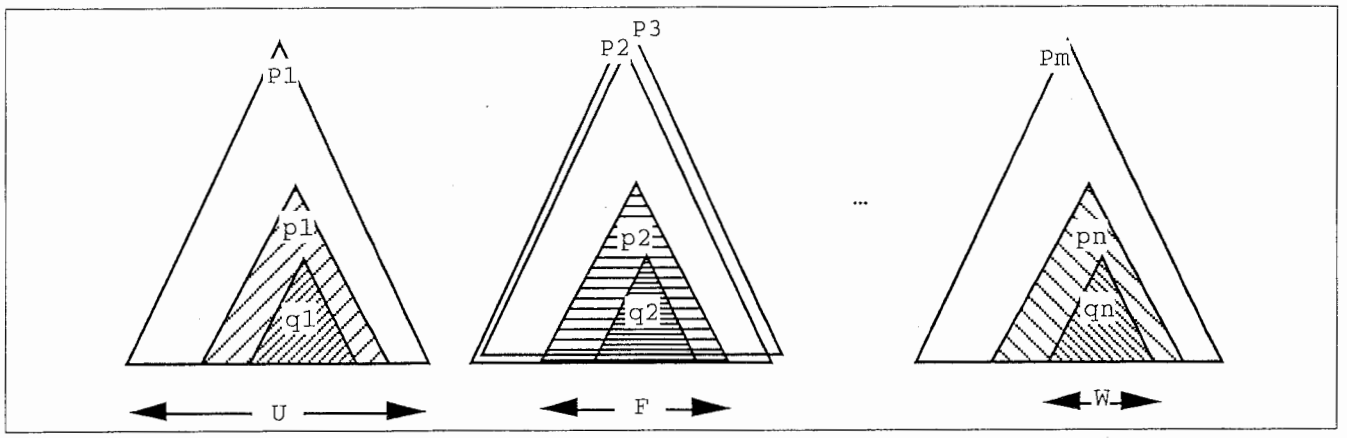
a. Ambiguity (occurrence)

An *ambiguity occurrence*, or simply *ambiguity*, A of multiplicity n (n≥2) relative to a representation system R, may be formally defined as:

$$A = (U, V, \langle P_1, P_2 \ldots P_m \rangle, \langle p_1, p_2 \ldots p_n \rangle), \text{ where } m \geq n \text{ and:}$$

- U is a complete utterance, called the *context* of the ambiguity.
- V is a fragment of U, usually, but not necessarily connex, the *scope* of the ambiguity.
- $P_1, P_2 \ldots P_m$ are all proper representations of U in R, and $p_1, p_2 \ldots p_n$ are the parts of them which represent V.
- For any fragment W of U strictly contained in V, if $q_1, q_2 \ldots q_n$ are the parts of $p_1, p_2 \ldots p_n$ corresponding to W, there is at least one pair $q_i, q_j$ $(i \neq j)$ such that $q_i = q_j$.

This may be illustrated by the following diagram, where we take the representations to be tree structures represented by triangles.




Here, $P_2$ and $P_3$ have the same part $p_2$ representing V, so that m>n.

b. Ambiguity kernel

The kernel of an ambiguity $A = (U, V, <P_1, P_2 \ldots P_m>, <p_1, p_2 \ldots p_n>)$ is the scope of A and its (proper) representations: $K(A) = (V, <p_1, p_2 \ldots p_n>)$.

In a data base, it will be enough to store only the kernels, and references to the kernels from the utterances.

## 4.4 Ambiguity type and ambiguity pattern

a. Ambiguity type

The *type* of A is the way in which the $p_i$ differ, and must be defined relative to each particular R.

If the representations are complex, the difference between 2 representations is defined recursively. For example, 2 decorated trees may differ in their geometry or not. If not, at least 2 correpsonding nodes must differ in their decorations.

Further refinements can be made only with respect to the intended interpretation of the representations. For example, anaphoric references and syntactic functions may be coded by the same formal kind of attribute-value pairs, but linguists usually consider them as different ambiguity types.

When we define ambiguity types, the linguistic intuition should be the main factor to consider, because it is the basis for any disambiguation method. For example, syntactic dependencies may be coded geometrically in one representation system, and with features in another, but disambiguating questions should be the same.

b. Ambiguity pattern

An *ambiguity pattern* is a schema with variables which can be instantiated to a (usually unbounded) set of ambiguity kernels.

Here is an ambiguity pattern of multiplicity 2 corresponding to the example above.

```
NP[ x1 NP[ x2 x3 ]], NP[ NP[ x1 x2] x3 ].
```

We don't elaborate, as ambiguity patterns are specific to a particular representation system and a particular analyzer.

# II. Attributes and values use in manual labelling

The proposed text processor oriented format for ambiguity labelling is a first version, resulting from several attempts by the second author to label transcriptions or spoken and multimodal dialogues.

We describe this format with the help of a classical context-free grammar, written in the font used here for our examples, and insert comments and explanations in the usual font.

## II.1 Top level (piece)

```
<labelled_piece>          ::= <labelled_text> | <labelled_dialogue>
<labelled_text>           ::= <text_header> <full_text> <labelled_paragraphs>
<text_header>             ::= 'LABELLED TEXT:' <text_name>
<text_name>               ::= <quoted_text>
<quoted_text>             ::= '"' <text> '"'
<full_text>               ::= <paragraph> [<parag_sep> <paragraph>]*
<paragraph>               ::= <text> [<unsure_paragraph_or_turn_sep> <text>]*
<unsure_paragraph_or_turn_sep> ::= '||?'
<labelled_dialogue>       ::= <dialogue_header> <full_dialogue> <labelled_turns>
<dialogue_header>         ::= 'LABELLED DIALOGUE:' <dialogue_name>
<dialogue_name>           ::= <quoted_text>
<full_dialogue>           ::= <turn> [<turn_sep> <turn>]*
<turn>                    ::= <T-text> [<unsure_p_or_t_sep> <T-text>]*
<T-text>                  ::= <speaker_code> ':' <text>
```

This means that the labelling begins by listing the text or the transcription of the dialogue, thereby indicating segmentation problems with the mark `"||?"`.

## II.2 Paragraph or turn level

### 2.1 Structure of the list and associated separators

The labelling continues with the next level of granularity, paragraphs or turns. The difference is that a turn begins with a speaker's code.

```
<labelled_paragraphs>     ::= <labelled_paragraph>+
<labelled_paragraph>      ::= <parag_text> <labelled_utterance>
                            |'PARAG' <parag_text> <labelled_utterances> ['/PARAG']
<parag_text>              ::= <utterance>[<utterance_sep> <utterance>]*
```

The mark PARAG must be used if there is more than one utterance. /PARAG is optional and should be inserted to close the list of utterances, that is if the next paragraph contains only one utterance and does not begin with PARAG. This kind of convention is inspired by SGML, and it might actually be a good idea in the future to write down this grammar in the SGML format.

```
<utterance>               ::= <text> [<unsure_utterance_sep> <text>]*
<unsure_utterance_sep>    ::= '|?'
<labelled_turns>          ::= <labelled_turn>+
<labelled_turn>           ::= <turn_text> <labelled_utterances>
                            |'TURN' <turn_text> <labelled_utterances> ['/TURN']
```

We use the same convention for TURN and /TURN as for PARAG and /PARAG.

```
<turn_text>               ::= <speaker_code> ':' <parag_text>
```

### 2.2 Representation of ambiguities of segmentation

If there is an ambiguity of segmentation in paragraphs or turns, there may be more labelled paragraphs or turns than in the source. For example, A ||? B ||? C may give rise to A-B||C and A||B-C, and not to A-B-C and A||B||C. Which combinations are possible should be determined by the person doing the labelling.

The same remark applies to utterances. Take one of the examples given at the beginning of this paper:

```
OK |? so go back and is this number three |? right there |? shall I wait
here for the bus?
```

This is an A |? B |? C |? D pattern, giving rise to 8 possibilities. If the labeller considers only the 4 possibilities A|B|C-D, A|B|C|D, A|B-C|D, and A-B-C|D, the following 7 utterances will be labelled:

```
A       OK
A-B-C   OK so go back and is this number three right there
B       so go back and is this number three
B-C     so go back and is this number three right there
C       right there
C-D     right there shall I wait here for the bus?
D       shall I wait here for the bus?
```

## II.3 Utterance level

### 3.1 Structure of the lists and associated separators

```
<labelled_utterances>   ::= <labelled_utterance>
                          | ['UTTERANCES'] <labelled_utterance>+
<labelled_utterance>    ::= <I-text> <ambiguity_kernels>
```

<I-text> means "indexed text": at the end of the scope of an ambiguity, we insert a reference to the corresponding ambiguity kernel, exactly as one inserts citation marks in a text.

### 3.2 Headers of ambiguity kernels

```
<ambiguity_kernels>     ::= <ambiguity_kernel>*
```

There may be no ambiguity in the utterance, hence the use of "*" instead of "+" as above.

```
<ambiguity_kernel>      ::= '(' <kernel_header> <ambiguity_labels> ')'
<kernel_header>         ::= 'ambiguity' <kernel_id> ['-' <TM_code>]
<kernel_id>             ::= <ref_piece> '-' <number> [']*
```

For example, a kernel header may be: `"ambiguity EMMI10a-2'-5.1"`. This is ambiguity kernel number 2' in dialogue EMMI 10a, noted here EMMI10a, and 5.1 is M. Tomokiyo's hierarchical code.

```
<ambiguity_labels>      ::= <obligatory_labels> <other_labels>
```

### 3.3 Obligatory labels

```
<obligatory_labels>     ::= <scope> {<status> <importance> <type>}
```

By {A B C}, we mean any permutation of ABC: we don't insist that the labeller follows a specific order, only that the obligatory labels come first, with the scope as very first.

#### a. Scope

```
<scope>                 ::= '(scope' <quoted_text> ')'
```

#### b. Status

```
<status>                ::= '(status' <status_value> ')'
<status_value>          ::= 'expert_system' | 'interpreter' | 'user'
```

The status expresses the kind of supplementary knowledge needed to reliably solve the considered ambiguity. If "expert_system" is given, and if a disambiguation strategy decides to solve this ambiguity interactively, it may ask: the expert system, if any; the interpreter, if any; or the user (speaker). If I is given, it means that an expert system of the generic task at hand could not be expected to solve the ambiguity.

c. Importance

```
<importance>          ::= '(importance' <importance_value> ')'
<importance_value>    ::= 'crucial' | 'important' |'not-important'
                        | 'negligible'
```

This expresses the impact of solving the ambiguity in the context of the intended task. An ambiguity of negation scope is often crucial, because it may lead to two opposed understanding, as in "A did not push B to annoy C" (did A push B or not?).

An ambiguity of attachment is often only important, as the corresponding meanings are not so different, and users may correct a wrong decision themselves. That is the case in the famous example "John saw Mary in the park with a telescope".

From Japanese into English, although the number is very often ambiguous, we may also very often consider it as "not-important".

"Negligible" ambiguities don't really put obstacles to the communication. For example, "bus" in English may be "autobus" (intra-town bus) or "autocar" (inter-town bus) in French, but either translation will almost always be perfectly understandable given the situation.

d. Type

```
<type>                ::= '(type' <type_name_value> ')'
<type_name_value>     ::= ('structure' | 'attachment') '(' <structure>+ ')'
                        | ('communication_act' | 'CA') '(' <comm_act>+ ')'
                        | ('class' | 'cat) '(' <morpho_syntactic_class>+ ')'
                        | 'meaning' '(' <definition>+ ')'
                        | <target_language> '(' <translation>+ ')'
                        | 'reference' <reference_value>
                        | 'address' '(' <addressee>+ ')'
                        | 'situation' <situation>
                        | 'mode' <mode>
                        | …
```

The linguists may define more types.

```
<structure>           ::= '<' (<text> | <structure>+) '>'
<comm_act>            ::= 'yes' | 'acknowledge' | 'yn-question'
                        | 'inform' | 'confirmation-question' | …
<morpho_syntactic_class> ::= 'N' | 'V' | 'Adj' | 'Adv' | …
<definition>          ::= <quoted_text>
<translation>         ::= <quoted_text>
<reference_value>     ::= <undefined_ref_value>
                        | '(' (<defined_ref_value>|<undefined_ref_value>)+ ')'
<undefined_ref_value> :: '*somebody' | '*something'
<defined_ref_value>   ::= <quoted_text>
<addressee>           ::= '*speaker' | '*hearer' | '*client' | '*agent'
                        | '*interpreter'
<situation>           ::= <quoted_text>
<mode>                ::= 'infinitive' | 'indicative' | 'conjunctive'
                        | 'imperative' | 'gerund'
```

### 3.4 Other labels

Other labels are not obligatory. Their list is to be completed in the future as more ambiguity labelling is performed.

```
<other_labels>          ::= [<disambiguation_scope> | <multimodality>…]*
<disambiguation_scope>  ::= 'definitive' | 'long_term' | 'short_term' | 'local'
<multimodality>         ::= 'multimodal'
                            (<multimodal_help> | '(' <multimodal_help>+ ')'
<multimodal_help>       ::= 'prosody' | 'pause' | 'pointing' | 'gesture'
                          | 'facial_expression' |…
```

## III. Example of a short dialogue

### III.1 Complete labelling in text processor oriented format

In this section, the numbers given between square brackets are not part of the format and have just been inserted for convenience of cross-reference.

#### 1.1 Text of the dialogue

LABELLED DIALOGUE: "EMMI 10a"

[1] A: Good morning conference office how can I help you
[2] AA: [ah] yes good morning could you tell me please how to get from Kyoto station to your conference center
[3] A: /ls/ [ah] yes (can you tell me) [ah] (you) you're going to the conference center today
[4] AA: yes I am to attend thi [uh] Second International Symposium {on} Interpreting Telecommunications
[5] A: {[o?]} OK n' where are you calling from right now
[6] AA: calling from Kyoto station
[7] A: /ls/ OK, you're at Kyoto station right now
[8] AA: {yes}
[9] A: {/breath/} and to get to the International Conference Center you can either travel by taxi bus or subway how would you like to go
[10] AA: I think subway sounds like the best way to me
[11] A: OK [ah] you wanna go by subway and you're at the station right now
[12] AA: yes
[13] A: OK so [ah] you'll want to get back on thi subway going north
[14] AA: [hmm]
[15] A: and you'll take the subway north to Sanjo station
[16] AA: OK
[17] A: /ls/ at Sanjo station you'll get off and change trains to thi Keihan Kyotsu line
[18] AA: [hmm]
[19] A: OK
[20] AA: I can get on thi Keihan train [u?] basically at the same location that I got off of the subway
[21] A: [ah] yes at Sanjo station you'll just change trains and get on (thi subway that) [ah] thi Keihan Kyotsu line [ah] which will take you close to thi International Conference Center
[22] AA: OK
[23] A: so [ah] from Kyoto station you take thi subway north to Sanjo sta{tion}
[24] AA: {(is)} by any chance is the bus more direct than this
[25] A: [ah] well the bus is probably a little easier
[26] AA: [hmm]
[27] A: [ahm] (it's) it's a little easier because you can take the bus right from the bus station across the street from where you are
[28] AA: OK
[29] A: and it will take you directly to thi International Conference Center

[30] AA: well 'f that'll take me directly there then it sounds like a better way to go [then] by subway
[31] A: OK [ah] let me give you directions on how to get to the bus station you're at Kyoto station now
[32] AA: ye{s}
[33] A: {i}f you take exit six {(that)}
[34] AA: {OK}
[35] A: that will take you outside and put you directly in front of thi bus station
[36] AA: which side of Kyoto station is that
[37] A: that's the north side of Kyoto station
[38] AA: OK
[39] A: [ah] you'll walk right across the street to the bus station and take bus number five
[40] AA: OK
[41] A: and bus number five will take you directly to thi International Conference Center it's a special bus
[42] AA: OK
[43] A: /ls/ [umm] {(and)}
[44] AA: {(how mu}ch) how much is the bus
[45] A: thi bus ride is five hundred yen
[46] AA: OK
[47] A: and it leaves every half hour
[48] AA: OK (ho?') how long does it take to reach the Conference Center
[49] A: it takes about fifteen minutes
[50] AA: OK
[51] A: OK
[52] AA: yes
[53] A: is there anything else I can help you with
[54] AA: no I think I can get there from here
[55] A: OK great OK
[56] AA: OK thank you
[57] A: (yo?') you're welcome have a good day
[58] AA: OK
[59] A: good bye
[60] AA: good bye

## 1.2 Turns

LABELLED TURNS OF DIALOGUE "EMMI 10a"

TURN
[1] AA: Good morning, conference office, I? How can I help you?

UTTERANCES
AA: Good morning, conference office(1)

(ambiguity EMMI10a-1-2.2.8.3 ((scope "conference office")
(status expert_system)
(address (*speaker *hearer))
(importance not-important)
(multimodal facial-expression)
(desambiguation_scope definitive)))

AA: How can I help you?

*/TURN is not necessary here because another TURN appears.*

TURN
[2] AA: [ah] yes, good morning. I Could you tell me please how to get from Kyoto station to your conference center?

*The labeller distinguishes here a sure segmentation into 2 utterances.*

```
UTTERANCES
AA:  [ah] yes(2), good morning.

(ambiguity EMMI10a-2-5.1 ((scope "yes")
(status  user)
(type CA (yes acknowledge))
(importance  crucial)
(multimodal prosody)))

AA:  Could you tell me please how to get from Kyoto station to your conference center(3)?

(ambiguity EMMI10a-3-2.2.2 ((scope "your conference center")
(status  user)
(type structure (  <<your conference><center>>
                   <<your><conference center>>))
(importance negligible)
(multimodal prosody)))
/TURN
```

*TURN is not necessary if there only one utterance with no ambiguity of segmentation.*

```
[3]   A:  /ls/ [ah] yes (type CAn you tell me) [ah] (you) you're going to the conference center
          today(4)

(ambiguity EMMI10a-4-5.2 ((scope "today")
(status  accidental)
(situation "the day they are speaking")
(importance negligible)
(multimodal  "built-in calendar on screen")))

[4]   AA: yes I am to(5) attend thi [uh] Second International Symposium {on}
Interpreting Telecommunications

(ambiguity EMMI10a-5-3.1.2 ((scope "am to")
(status  user)
(type Japanese ("ねばならない"  "ことになっている"  "はずだ"))
(importance  important)))

[5]   A:  {[o?]} OK n' where are you calling(6) from right now(7)

(ambiguity EMMI10a-6-3.1.2 ((scope "calling")
(status  expert_system)
(type Japanese (``電話する" ``呼ぶ"▩
(importance  crucial)))

(ambiguity EMMI10a-7-2.1((scope "calling from right now")
(status  user)
(type structure (  <<calling from> <right now>>
                   <calling <from <right now>>>)
(importance  crucial)
(multimodal prosody)))

[6]   AA :  calling from Kyoto station

[7]   AA: /ls/ OK, you're at Kyoto station(8) right now.

(ambiguity EMMI10a-8-5.1 ((scope "you're at Kyoto station")
(status  expert_system)
(type CA (yn-question inform))
```

(importance crucial)
(multimodal prosody)))

[8] AA : {yes}

TURN
[9] A: {/breath/} and to get to the International Conference Center you can either travel by taxi bus or subway. I how would you like to go

UTTERANCES
A: {/breath/} and to get to the International Conference Center you can(9) either travel(9', 9") by taxi bus or subway(10).

(ambiguity EMMI10a-9-2.1 ((scope "can")
(status expert_system)
(type class (verb modal_verb))
(importance crucial)))

(ambiguity EMMI10a-9'-2.1 ((scope "the International Conference Center you can either travel")
(status expert_system)
(type structure ( <<<the International Conference Center> <you can>> <either travel>>
<<the International Conference Center> <you can <either travel>>>>)
(importance crucial)
(multimodal prosody)))

(ambiguity EMMI10a-9"-2.1 ((scope "travel")
(status expert_system)
(mode (infinitive imperative))
(importance crucial)))

(ambiguity EMMI10a-10-2.2.2 ((scope "taxi bus or subway")
(status expert_system)
(type structure (<taxi or bus or subway> <taxi-bus or subway>))
(importance important)
(multimodal prosody)))

A: How would you like to go
/TURN

*This example is of the same kind as the very famous one: "Time flies like an arrow"!*
*"Linguist's examples" are often derided, but they really appear in texts and dialogues.*
*However, as soon as they are taken out of context, they look again as artificial as "linguistst's examples!*

[10] AA: I think subway sounds(10) like(11) the best way to me

(ambiguity EMMI10a-10-3.1.1 ((scope "sounds")
(status interpreter)
(type CAt (verb noun))
(importance crucial)
(multimodal (prosody pause)))

(ambiguity EMMI10a-11-3.1.1 ((scope "like")
(status interpreter)
(type CAt (verb preposition))
(importance crucial)
(multimodal (prosody pause)))

[11] A: OK, [ah] you wanna go by subway and you're at the station right now(12).

(ambiguity EMMI10a-12-5.1 ((scope "you wanna go by subway and you're at the station right now")

(status expert-system)
(type CA (yn-question inform))
(importance crucial)
(multimodal prosody)))

[12] AA: yes

[13] A: OK so [ah] you'll want to(13) get back on thi subway going north(14)

(ambiguity EMMI10a-13-3.1.2 ((scope "want to")
(status interpreter)
(type Japanese ("たい" "べき"))
(type French ("vouloir" "devoir"))
(importance important)))

(ambiguity EMMI10a-14-2.2.2 ((scope "get back on thi subway going north")
(status user)
(type structure ( <get back <on thi subway> <going north>>
<get back <on thi subway <going north>>>))
(importance important)
(multimodal prosody)))

[14] AA: [hmm]

[15] A: and you'll take the subway north to Sanjo station

[16] AA: OK

[17] A: /ls/ at Sanjo station you'll get off(15) and change trains to thi Keihan Kyotsu line

(ambiguity EMMI10a-15-5.2 ((scope "get off and change trains")
(status user)
(type structure ( <<get off and change> trains>
<<get off> <and change trains>>))
(importance negligible)
(multimodal pause)))

[18] AA: [hmm]

[19] A: OK

[20] A: I can get on thi Keihan train [u?] basically at the same location that I got off of the subway(16)

(ambiguity EMMI10a-16-5.1 ((scope "I can get on...the subway.")
(status interpreter)
(type CA (inform confirmation-question yn-question))
(importance crucial)
(multimodal prosody)))

[21] AA: [ah] yes at Sanjo station you'll just change trains and get on (thi subway that) [ah] thi Keihan Kyotsu line [ah] which will take you close to thi International Conference Center

[22] AA: OK

[23] A: OK so [ah] you'll want to get back on thi subway going north(14)

[24] AA: {(is)} by any chance is the bus more direct than this

[25] A: [ah] well the bus is probably a little easier

[26] AA: [hmm]

[27] A: [ahm] (it's) it'(17)s a little easier because you can take the bus right(18) from the bus station across the street from where you are

```
(ambiguity EMMI10a-17-2.2.7.1 ((scope "it")
(status expert_system)
(type reference ("bus" *something))
(importance not-important)))

(ambiguity EMMI10a-18-3.1.1 ((scope "right")
(status expert_system)
(type meaning ("right of right/left" "properly" "exactly" "straightly"))
(type Japanese ("右に" "適切に" "正確に" "まっすぐに"))
(type French ("a droite" "correctement" "juste" "tout droit"))
(importance important)))
```

[28] AA: OK

[29] A: and it(19) will take you directly to thi International

```
(ambiguity EMMI10a-19-2.2.7.1 ((scope "it")
(status expert_system)
(type reference ("bus" *something))
(importance not-important)))
```

*This kernel is the same as (17), but the labeller is not forced to check for identity. The data base should take care of that later.*

[30] AA: well 'f that'(20)ll take me directly there then it(19) sounds(21) like a better way to go then by subway Conference Center

```
(ambiguity EMMI10a-20-2.2.7.1 ((scope "that")
(status interpreter)
(type reference ("bus" *something))
(importance not-important)))

(ambiguity EMMI10a-21-2.2.7.1 ((scope "sounds")
(status interpreter)
(type meaning ("seem" "make a noise"))
(importance crucial)))
```

TURN

[31] A: OK [ah] let me give you directions on how to get to the bus station. | You're at Kyoto station now.

UTTERANCES

A: OK [ah] let me give you directions on how to get to the bus station.

A: You're at Kyoto station now(22)

```
(ambiguity EMMI10a-22-5.1 ((scope "you're at Kyoto station now")
(status interpreter)
(type CA (inform confirmation-question yn-question))
(importance crucial)
(multimodal prosody)))
```

[32] AA: ye{s}

[33] A: {i}f you take exit six {(that)}

[34] AA: {OK}

[35] A: that(23) will take you outside and put you directly in front of thi bus station

```
(ambiguity EMMI10a-23-2.2.7.1 ((scope "that")
(status interpreter)
(type reference ("exit six" *something))
(importance not-important)))
```

[36] AA: which side of Kyoto station is that(23)

[37] A: that(23)s the north side of Kyoto station

[38] AA: OK

[39] A: [ah] you'll walk right(18) across the street to the bus station and take bus number five(24)

```
(ambiguity EMMI10a-24-2.2.2 ((scope "bus number five")
(status expert_system)
(type structure ( <<bus number><five>>
                  <<bus> <number five>>))
(importance negligible)
(disambiguation_scope definitive)
(multimodal prosody)))
```

[40] AA: {OK}

[41] A: and bus number five(24) will take you directly to thi International Conference Center it'(25)s a special bus

```
(ambiguity EMMI10a-25-2.2.7.1 ((scope "it")
(status interpreter)
(type reference ("bus" "International Conference Center" *something))
(importance important)))
```

[42] AA: OK

[43] A: /ls/ [umm] {(and)}

[44] AA: {(how much)} how much is the bus(26)

```
(ambiguity EMMI10a-26-5.2 ((scope "bus")
(status accidental)
(ellipsis ("bus fare" "bus as vehicle"))
(importance negligible)))
```

[45] A: thi bus ride is five hundred yen

[46] AA: OK

[47] A: and it(27) leaves every half hour

```
(ambiguity EMMI10a-27-2.2.7.1 ((scope "it")
(status expert_system)
(type reference ("bus" *something))
(importance important)))
```

[48] AA: OK (ho?') how long does it take to reach the Conference Center

[49] A: it takes about fifteen minutes

[50] AA: OK

[51] A: OK

[52] AA: yes

[53] A: is there anything else I can help you with

[54] AA: no I think I can get there(28) from here(29)

```
(ambiguity EMMI10a-28-2.2.7.3 ((scope "there")
(status accidental)
(deictic ("hearer's place" *somewhere))
(importance negligible)))

(ambiguity EMMI10a-29-2.2.7.3 ((scope "here")
(status intepreter)
(deictic ("speaker's place" somewhere))
(importance negligible)))
```

[55] A: OK great OK

[56] AA: OK thank you

[57] A: (yo?') you're welcome have a good day

[58] AA: OK

[59] A: good bye

[60] AA: good bye

### III.2 Fragment in a data base oriented format

The idea is simply to use a line-oriented format, each line beginning with a keyword coresponding to the part being labelled. If the information does not fit on one line, the keyword is repeated at the beginning of the next line.

The following fragment (turns 1—7) ilustrates the idea. The main point is that such a format is easier to handle by traditional DBMS systems. The details of the formats may vary, but it is always required that translation from one format into the other is possible, without loss of information.

2.1 Text of the dialogue

```
HEADING:        LABELLED DIALOGUE: "EMMI 10a"

TEXT:           A:  Good morning, conference office, how can I help you?
TEXT:           A:  Good morning conference office how can I help you
TEXT:           AA: [ah] yes good morning could you tell me please how to get from
TEXT:           Kyoto station to your conference center
TEXT:           A:  /Is/ [ah] yes (can you tell me) [ah] (you) you're going to the
TEXT:           conference center today
TEXT:           AA: yes I am to attend thi [uh] Second International Symposium {on}
TEXT:           Interpreting Telecommunications
TEXT:           A:  {[o?]} OK n' where are you calling from right now
TEXT:           AA: calling from Kyoto station
TEXT:           A:  /Is/ OK you're at Kyoto station right now
```

## 1.2 Turns

Blank lines have been inserted only to make reading easier.

```
TURNS:                  LABELLED TURNS OF DIALOGUE "EMMI 10a"

TURN:                   [1]
TEXT:                   AA: Good morning, conference office, |? How can I help you?

UTTERANCE:              [1.1]
TEXT:                   Good morning, conference office(1)

AMBIGUITY:              EMMI10a-1-2.2.8.3
SCOPE:                  "conference office"
STATUS:                 expert_system
ADDRESS:                (*speaker *hearer)
IMPORTANCE:             not-important
MULTIMODAL:             facial-expression
DISAMBIGUATION_SCOPE:   definitive

UTTERANCE:              [1.2]
TEXT:                   AA: How can I help you?

TURN:                   [2]
TEXT:                   AA: [ah] yes, good morning. I Could you tell me please how to get
TEXT:                   from Kyoto station to your conference center?
COMMENT:                Sure segmentation into 2 utterances.

UTTERANCE:              [2.1]
TEXT:                   AA: [ah] yes(2), good morning.

AMBIGUITY:              EMMI10a-2-5.1
SCOPE:                  "yes"
STATUS:                 accidental
TYPE:                   CA (yes acknowledge)
IMPORTANCE:             crucial
MULTIMODAL:             prosody

UTTERANCE:              [2.2]
TEXT:                   AA: Could you tell me please how to get from Kyoto station to your
TEXT:                   conference center(3)?

AMBIGUITY:              EMMI10a-3-2.2.2
SCOPE:                  "your conference center"
STATUS:                 user
TYPE:                   structure ( <<your conference><center>>
TYPE:                             <<your><conference center>>)
IMPORTANCE:             negligible
MULTIMODAL:             prosody

TURN:                   [3]
TEXT:                   A:  /ls/ [ah] yes TYPE:  CAn you tell me) [ah] (you) you're going
TEXT:                   to the conference center today(4)

AMBIGUITY:              EMMI10a-4-5.2
SCOPE:                  "today"
STATUS:                 accidental
SITUATION:              "the day they are speaking"
IMPORTANCE:             negligible
MULTIMODAL:             "built-in calendar on screen"
```

```
TURN:           [4]
TEXT:           AA: yes I am to(5) attend thi [uh] Second International Symposium
TEXT:           {on} Interpreting Telecommunications

AMBIGUITY:      EMMI10a-5-3.1.2
SCOPE:          "am to"
STATUS:         user
TYPE:           Japanese ("ねばならない"  "ことになっている"  "はずだ")
IMPORTANCE:     important

TURN:           [5]
TEXT:           A:  {[o?]} OK n' where are you calling(6) from right now(7)

AMBIGUITY:      EMMI10a-6-3.1.2
SCOPE:          "calling"
STATUS:         expert_system
TYPE:           Japanese ("電話する" "呼ぶ")
IMPORTANCE:     crucial

AMBIGUITY:      EMMI10a-7-2.1
SCOPE:          "calling from right now"
STATUS:         user
TYPE:           structure ( <<calling from> <right now>>
TYPE:                       <calling <from <right now>>>)
IMPORTANCE:     crucial
MULTIMODAL:     prosody

TURN:           [6]
TEXT:           AA :        calling from Kyoto station

TURN:           [7]
TEXT:           AA: /ls/ OK, you're at Kyoto station(8) right now.

AMBIGUITY:      EMMI10a-8-5.1
SCOPE:          "you're at Kyoto station"
STATUS:         expert_system
TYPE:           CA (yn-question inform)
IMPORTANCE:     crucial
MULTIMODAL:     prosody
```

## Conclusion

Although many studies on ambiguities have been published, the specific goal of studying ambiguities in the perspective of interactive disambiguation in automated text and speech translation systems has led us to explore some new ground and to propose the new concept of "ambiguity labelling". Several dialogues from EMMI-1 [11] and EMMI-2 [20] have already labelled (in Japanese and English). Attempts have also been made on French texts and dialogues. In the near future, we hope to refine our ambiguity labelling, and to label WOZ dialogues from EMMI-3 [22]. In parallel, the specification of MIDDIM-DB, a HyperCard based support for the ambiguity data base under construction, is being reshaped to implement the new notions introduced here: ambiguity kernels, occurrences, and types.

## Acknowledgments

We are very grateful to Dr. Y. Yamazaki, president of ATR-ITL, Mr. T. Morimoto, head of Department 4, and Dr. K.-H. Loken-Kim, for their constant support to this project, which one of the projects funded by CNRS and ATR in the context of a memorandum of understanding on scientific cooperation. Thanks should also go to M. Axtmeyer, L. Fais and H. Blanchon, who have contributed to the study of ambiguities in real texts and dialogues, and to M. Kurihara, for his programming skills.

## References (chronological)


[1] **Boitet C. (1989)** *Speech Synthesis and Dialogue Based Machine Translation.* Proc. ATR Symp. on Basic Research for Telephone Interpretation, Kyoto, December 1989, 6-5-1:22.

[2] **Maruyama H., Watanabe H. & Ogino S. (1990)** *An Interactive Japanese Parser for Machine Translation.* Proc. COLING-90, Helsinki, 20-25/8/90, H. Karlgren, ed., ACL, vol. 2/3, 257-262.

[3] **Boitet C. & Blanchon H. (1993)** *Dialogue-based MT for monolingual authors and the LIDIA project.* Rapport de Recherche, RR-918-I, IMAG, GETA, UJF&CNRS, Grenoble, mai 1993.

[4] **Boitet C. (1993)** *Practical Speech Translation Systems will integrate human expertise, multimodal communication, and interactive disambiguation.* Proc. MTS-IV, Kobe, 18–22 July 1993, 173–176.

[5] **Blanchon H. (1993)** *Report on a stay at ATR.* Project report (MIDDIM), GETA & ATR-ITL, July 1993, 30 p.

[6] **Boitet C. (1993)** *Human-Oriented Design and Human-Machine-Human Interactions in Machine Interpretation.* Technical Report, ATR Interpreting Telecommunications, Aug. 93, 13 p.

[7] **Boitet C. (1993)** *Multimodal Interactive Disambiguation: first report on the MIDDIM project.* Technical Report, ATR Interpreting Telecommunications, Aug. 93, 16 p.

[8] **Black E., Garside R. & Leech G. (1993)** *Statistically-Driven Grammars of English: the IBM/Lancaster Approach.* J. Aarts & W. Mejs, ed., Language and Computers: Studies in Practical Linguistics, Rodopi, Amsterdam, 248 p.

[9] **Boitet C. & Loken-Kim K.-H. (1993)** *Human-Machine-Human Interactions in Interpreting Telecommunications.* Proc. International Symposium on Spoken Dialogue, Tokyo, 10–12 November 1993, Waseda University, 4 p..

[10] **Boitet C. & Loken-Kim K.-H. (1993)** *Human-Machine-Human Interactions in Interpreting Telecommunications.* Proc. International Symposium on Spoken Dialogue, Tokyo, 10–12 November 1993, Waseda University, 4 p..

[11] **ATR-ITL (1994)** *Transcriptions of English Oral Dialogues Collected by ATR-ITL using EMMI (from TR-IT-0029, ATR-ITL).* EMMI report, ATR-ITL, January 1994, 33 p. (file edited at GETA)

[12] **Boitet C. & Axtmeyer M. (1994)** *Documents prepared for inclusion in MIDDIM-DB.* Internal report, GETA, IMAG (UJF & CNRS), June 1994, 3 p.

[13] **Axtmeyer M. (1994)** *Analysis of ambiguities in a written abstract (MIDDIM project).* Internal report, GETA, IMAG (UJF & CNRS), July 1994, 17 p.

[14] **Winship J. (1994)** *Building MIDDIM-DB, a HyperCard data-base of ambiguities and disambiguation methods.* ERASMUS project report, GETA, IMAG (UJF & CNRS) & University of Sussex at Brighton, 22 July 1994, 22 p.

[15] **Blanchon H. (1994)** *Perspectives of DBMT for monolingual authors on the basis of LIDIA-1, an implemented mockup.* Proc. 15th International Conference on Computational Linguistics, COLING-94, Kyoto, Japan, 5-9 Aug. 1994, vol. 1/2, 115—119.

[16] **Tomokiyo M. & Loken-Kim K.-H. (1994)** *Ambiguity analysis and MIDDIM-DB.* MIDDIM Report, ATR-ITL & GETA-IMAG, Aug. 94, 74 p.

[17] **Blanchon H. (1994)** *Pattern-based approach to interactive disambiguation: first definition and experimentation.* Technical Report, ATR-ITL, Sept. 94.

[18] **Boitet C. (1994)** *On the design of MIDDIM-DB, a data base of ambiguities and disambiguation methods.* MIDDIM Report, ATR-ITL & GETA-IMAG, Sept. 94, 12 p.

[19] **Tomokiyo M. (1994)** *Ambiguity Classification and Representation.* Proc. Natural Language Understanding and Models of Communication, Oct. 94.

[20] **Park Y.-D. & Loken-Kim K.-H. (1994)** *Text Database of the Telephone and Multimedia Multimodal Interpretation Experiment.* Technical Report, ATR-ITL, Dec. 94, 161 p.

[21] **Park Y.-D., Loken-Kim K.-H. & Fais L. (1994)** *An Experiment for telephone versus multimedia multimodal Interpretation: Methods and Subject's Behavior.* Technical Report, ATR-ITL, Dec. 94, 15 p.

[22] **Park Y.-D., Loken-Kim K.-H., Mizunashi S. & Fais L. (1995)** *Transcription of the Collected Dialogue in a Telephone and Multimedia/Multimodal WOZ Experiment.* Technical Report, ATR-ITL, Feb. 95, 123 p.

[23] **Boitet C. & Blanchon H. (1995)** *Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mockup.* Machine Translation. (to appear)


-o-o-o-o-o-o-o-o-o-o-o-