Categorial grammars used to partial parsing of spoken language

Jeanne Villaneau[1] Jean-Yves Antoine[2]

Valoria, Université de Bretagne Sud, Vannes-Lorient, France

Abstract

Spoken language understanding is a challenge for the development of Spoken Dialogue Systems. Recognition errors and speech repairs make it impossible to get complete syntactic analysis. Shallow parsing and chunking seem to be efficient in order to start both a robust and precise analysis.

This paper describes experiments made with LOGUS, a spoken understanding system based on incremental methology. It presents the first step of the parsing, a chunking based on rules of categorial grammars and pregroups. These formalisms are very appropriate for this treatment and we argue they could be more widely used for applications of this type.

Key words: speech understanding, chunking, incremental analysis, categorial grammars

---

1 Natural Language Understanding in Spoken Dialogue Systems

Over about the last ten years, progresses in speech recognition have made possible the development of spoken dialogue systems. If a great number of these systems are still experimental prototypes, some of them have given rise to commercial applications, thus showing the advances of the domain. However, the intended tasks these systems achieve are very precise and limited to a very constrained domain: for example, the system can give train or air table information (Lamel 2000), weather information (Zue 2000), etc. Moreover, dialogue is very often machine-directed and give little flexibility to the users in expressing their queries.

---

[1] Jeanne.Villaneau@univ-ubs.fr

[2] Jean-Yves.Antoine@univ-ubs.fr

Spoken language understanding is an essential element for these systems, whose architecture is generally similar to the one showed in figure 1. An automatic speech recognizer performs speech-to-text conversion. Lower down, the module of natural language understanding (NLU) builds a semantic representation of the utterance. This representation is used by the dialogue manager; it decides on the responses to give to the user or on the questions to ask him in order to specify his request.

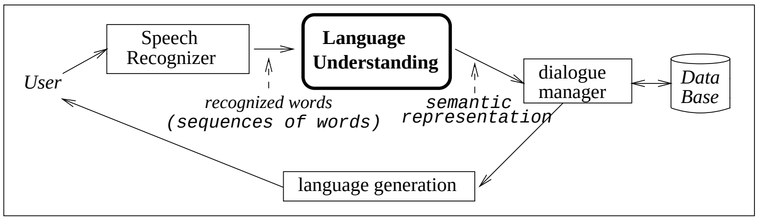

Fig. 1. Architecture of a spoken dialogue system

Because implicit knowledges and ambiguities are always present in natural language, automatic understanding is a very difficult problem. For spoken language understanding, recognition errors and spontaneous speech artifacts increase the difficulty:

- Although the use of statistic methods has led speech recognition to make outstanding advances during the past decade, the errors caused by the latter are - and seem doomed to remain - frequent. Recognition error rates on the words are from 20 to 30% in the application conditions relating to an answering service: spontaneous speech, multiplicity of the users, background noise, real time. Recognition errors break the syntactic and semantic structure of the utterance and it seems that they don't follow any rules. In the statistical language models used in the speech recognizers, the context of a word is defined at best as the two previous words: these models can't check syntactic or semantic coherence of the recognition proposal.
- Speech "repairs" are another source of problems, specific to spontaneous speech understanding. This generic term of "repairs" refers to hesitations, repetitions and autocorrections of the speaker looking for his words: for example, *"il y a euh quels sont les restaurants dans dans le coin" ("there is euh what are the restaurants around around here").* Like recognition errors, repairs break syntactic regularity of the utterance and disturb its linear development. Furthermore, the scope of these repairs is very variable: the speaker can correct the beginning of a word, the beginning of a syntagm, a full syntagm or a sentence. However, they seem to follow some rules: studies have been made on repairs relating to prepositional nominal syntagms in French. They have showed that the alteration of the repair - the speech that the speaker intends as replacement - contains the full syntagm with a frequence of 70% (Martinie 2001): *"dans ce dans cet hôtel" ("in this in*

this hotel"). So, minimal syntactic structures are preserved: "good" function words remain near their related content word.

In the face of such problems, a large variety of methods have been explored. All currently operational systems use the very constrained nature of the considered task. Very often, the simplicity of this task allows one to build semantic frames in order to represent all possible queries. So, understanding can be reduced to detection of keywords or phrases in order to instanciate the different parameters of these patterns. For example, in the MASK system, the task consists in giving train table information. Only the underlined words are used to understand the query: *" je veux aller demain matin de Paris à Marseille en passant* par Lyon*" (I want to go tomorrow morning from Paris to Marseille via Lyon).* The phrase *"de Paris"* is interpreted without ambiguity as a city-departure and full linguistic analysis is not necessary.

These methods are called *selective methods*; they are robust and effective if the system has to perform a specific action. Because the frames give the structure of the queries, frame-based approaches make parsing possible without full linguistic analysis. On the other hand, it is no sure that these approaches are sufficient if the domain becomes less constrained, if interaction between the system and the user is wanted or if the system must understand less simple requests.

2 Partial parsing and spontaneous spoken language

A no-frame based understanding must use linguistic analysis in order to build a semantic structure which semantically represents the utterance. This section and the following one explain why LOGUS system uses an incremental parsing whose first step is a chunking and why categorial grammars are used in this step.

Most formalisms used for natural language processing are based on assumption of syntactic correction of the valid parsed utterances. For the previously mentionned reasons, this assumption is not tenable for parsing of spoken utterances. Various processes can be used in order to try to bypass these difficulties: for example, in TINA, the understanding system of the MIT, a selective method is used for the utterances for which a full parsing has failed (Seneff 1992). This second processing is totally dissociated from the first one and the ratio of the utterances subjected to this treatment is unknown. Another attempt was tested by Bear and his colleagues: it consists in detecting and correcting speech repairs before parsing. However, repairs are *"easily confused both with false positives and with other repairs"* (Bear 1992). This automatic detection is difficult and the results are not convincing.

Otherwise, for a few years, partial syntactic analysis have been developed for the design of robust parsing systems. They are designed for engineering, for example in order to extract information from electronic corpora (Aït-Mokhtar 2002). The qualities required for these tasks are similar to the qualities required for spoken language understanding in an answering system: efficiency, robustness and quickness. The final purpose is not to check or to study grammaticality of the parsed utterance but to draw its general meaning. The basic principle is to produce minimal syntactic constituents, often called *chunks* (Abney 1991). These syntactic groups are composed of a lexical head to which local dependent terms are linked. The treatments used to obtain this segmentation are based on simple formalisms. They can form the first step of an incremental parsing whose next steps are meant to link the obtained components. The observations showing that the repairs generally preserve minimal syntactic structures can lead to thinking that a step of *chunking* can be used in order to parse spontaneous spoken language. The following section presents the LOGUS system where this methodology is used, with support of categorial grammars.

3 Architecture and basic principles of LOGUS system

LOGUS[3] is a system designed for spontaneous French spoken language understanding in human-computer dialogue; it is relevant to a delimited domain but noticeably wider than the generally considered domains: the understanding cannot be frame-based but a semantic knowledge of the application domain can be used. Parsing must be robust in order to resist the features of spontaneous spoken language but it must also be precise in order to remove ambiguities and to accurately detect the intentions of the speaker. In order to reach these objectives, two general principles are applied: using an incremental methodology of parsing and combining syntactic and semantic criteria.

The test-domain is touristic information: it is wide enough to require the use of relatively complex structures: *"le tarif des chambres doubles et simples au Caumartin ou au Crillon" (the price for double and single rooms in Caumartin or Crillon)* yet well delimited. Despite the use of a lexicon and an ontology closely related to the domain test, the system can be used for another application, without more changes. The first subsection describes target language used for the semantic representations. The following ones present the method used in order to build these representations.

---

[3] LOGUS : LOGical Understanding System

3.1 Logic approach - Target language

The semantic representations the NLU builds are used by the dialogue manager. The choice for the target language must take this objective into account. The general intention of the user is known: information query or task order. On the other hand, intentions expressed in the utterances during the dialogue can be various. For example, for a single information request, they can be partial or total confirmations or rejects, precisions, etc. You need to detect them correctly for the good development of the dialogue.

The sentences exchanged during a human-computer dialogue in order to make a precise task fall in the scope of the speech-acts theory that Austin (Austin 1970) and Searle (Searle 1970) have initiated: speech is used *"to do things"*. Illocutory logic of D. Vanderveken (Vanderveken 2001) provides formalism in order to take this speech pragmatics into account: the representation of a sentence is the application of an illocutory force to a propositional content. In the perspective of logical dialogue managing, the semantic representation LOGUS provides is a formula directly inspired by this formalism. Objects and properties related to application domain and detected in the parsed sentence are linked to build a structure called *object string*. It plays the role of propositional content. The logic formula is obtained by application of a language act to this structure. Object string and language act are built with concepts and conceptual relations in order to enable the logic formula to be convertible into a conceptual graph (Sowa 1984).

Figure 2 gives an example of semantic representation. The concepts (rectangular boxes) *list*, *hotel*, *art_gallery* are object labels, (*not expensive*) and (*name "Louvre"*) are property determinations. Conceptual relations (pseudo-elliptical boxes) *cost*, *near*, *identity* are property labels. The conceptual relation *of* expresses a subordination relation between objects. In the final formula, each object is represented by a label and the list of its properties.

3.2 Steps of parsing

Parsing is incremental and bottom-up. The final formula is obtained with gradual composition of the elements contained in the utterance and known by the system. Successive steps use formalisms adapted from classic syntactic formalisms. These adaptations are meant to combine syntactic and semantic arguments. Syntactic constraints are gradually weakened during the parsing.

General architecture of the system is given in figure 3. The parsing begins with *chunking* by which utterance is divided in syntactic minimal structures. The formalism used in this step derives from categorial grammars. Only this

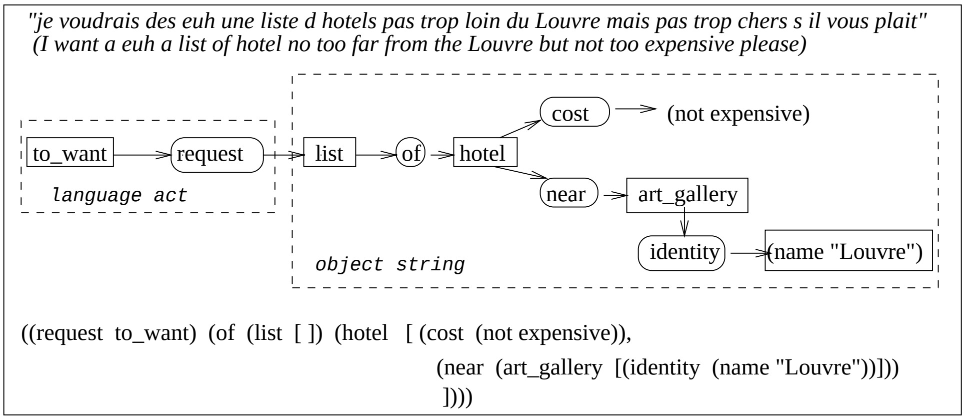

Fig. 2. Example of semantic representation

first step is decribed in this report. The following step consists in establishing dependency links between the *chunks*: it is based on application of rewriting syntactic-semantic rules. These rules are applied by levels, according to the applied syntactic constraints. The last step is an interpretation of the utterance using dialogue context in order to solve references and to complete language acts and object strings.

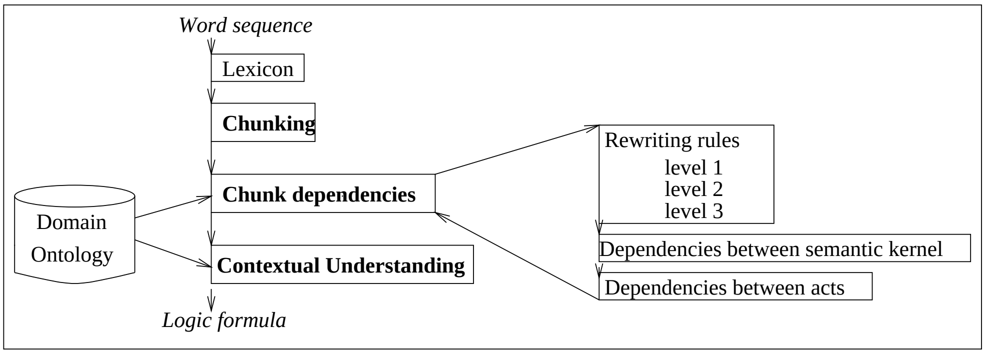

Fig. 3. Architecture of Logus system

3.3 Constituents representation

Formalism used to represent constituents during the parsing is meant to combine syntactic and semantic approaches. At the same time, it is meant to preserve genericity of the used rules. Particularly, the concepts specific to the application domain should not appear in these rules. The choosen answer is to represent constituents with a triplet $< C,\ R,\ T >$ where

- C is the *syntactic category* of the constituent. It plays the role of a category

during the *chunking*. In the following steps, it is used to express syntactic constraints.
- R is the semantic role. It points out semantic function of the constituent in the application (object, property, etc.). Its roles are similar to the roles of the syntactic category.
- T is the *semantic translation*. It belongs to target language and it is closely related to the application domain.

The applied principle is that the rules are supported by the two first elements of the triplets. So, these rules are generic; the system may be seen as an interpreter of knowledges included in a program made with the lexicon and the ontology.

4 Logus **chunking**

4.1 Minimal chunks

Chunks are defined as *"syntagmatic un-recursive units."*. This flexible definition can produce various segmentations. Relating to Logus, first evaluations of the system have led to a very narrow scope of the *chunks*: recognition errors make it dangerous to link objects or properties from syntactic criteria, without checking these links with the domain ontology. For example, the occurrence of the word *"doubles"* in the expression *"les horaires doubles d'ouverture du Louvre" (the double hours of opening of Louvre)* comes from a recognition error, because of an hesitation over the word *"d'ouverture"*[4]. A standard chunking links the adjective *"doubles"* with the preceding common noun *"horaires"*[5], and leads to a semantically uninterpretable link. So, the notion of *chunks* in Logus is restricted by the two definitions and the rule given below:

Definition 1: A lexeme is *lexical* if its semantic component includes an object, a property or the expression of a language act defined in the application.

Definition 2: A *minimal chunk* includes at the most[6] one lexical lexeme.

Rule: The chunking builds only *minimal chunks*.

This definition changes the notion of *chunk* itself. Whereas usually the determining of a *chunk* is based on exclusively syntactic criteria, the definition of

---

[4] Error observed with the software of vocal dictation ViaVoice (IBM).

[5] In French, the adjective *"doubles"* is usually located behind the common noun it qualifies.

[6] Subordination and coordination marks form *chunks* without lexical lexeme.

minimal chunk includes a semantic criterion. This choice is representative of a general principle: in LOGUS, syntax and sematics interpenetrate throughout the analysis. However, these constraints on the *chunks* lead to pretty unorthodox segmentations: *[dans deux] [ou] [trois] [heures] (in two or three hours)*, *[une chambre] [double] (a double room)*, etc.

4.2 Segmentation rules

4.2.1 AB grammars

The local links used to build the *chunks* are generated by an algebric grammar: a simple formalism is enough for their determination. In LOGUS, the choice is Categorial Grammars of AB type (Bar-Hillel 1964) for two reasons:

(1) A general principle is a bottom-up analysis. The reconstruction of the most probable meaning of the parsed utterance is made from the meaning of the words. So it is advisable to choose a lexicalized formalism.
(2) Otherwise, categorial grammars enable you to make a direct link between grammar rule application and building of the semantic formula, through composition of λ-terms.

In LOGUS, the notion of category is extended to the two first elements of the definition triplets. Fractional categories are mainly given to function words. In the related definitions, the semantic translation is an abstraction (in the sense of λ-terms). For example, in one of its definitions, the word *"le" (the)* has $(gn\ nomc\ (det\ def\ sing))/nomc$ as a syntactic category and $object/object$ as a semantic role. The abstraction related to this definition is identity $(\lambda x.x)$. So, it can be linked to a common noun application object located on its right.

The two rules of categorial grammars of AB type: $(A,\ (A\backslash B) \rightarrow B)$ and $(B/A,\ A) \rightarrow B$ are extended to the two first elements of the definition triplets. The semantic translation of the resulting triplet is obtained by applying the semantic translation abstraction of the "fractional" triplet to the semantic translation of the "atomic triplet". The links between words in a *chunk* are obtained by applying the two following rules:

$$< C_A,\ R_A,\ S_A >,\ < C_A\backslash\ C_B,\ R_A\backslash\ R_B,\ F > \rightarrow < C_B,\ R_B,\ (F\ \ S_A) >$$

$$< C_B/C_A,\ R_B\ /R_A,\ F >,\ < C_A,\ R_A,\ S_A > \rightarrow < C_B,\ R_B,\ (F\ \ S_A) >$$

In the following example, the second rule is applied twice (*g_adj* is for *"adjectival group"* and *prop* means *property*):

"pas" (not)	$< (g\_adj/adjective),$	$(prop\ R)/(prop\ R),$	$(\lambda x.(not\ x)) >$
"trop" (too)	$< (adjective/adjective),$	$(prop\ R)/(prop\ R),$	$\lambda x.x >$
"cher" (expensive)	$< adjective,$	$(prop\ cost),$	$expensive >$
"pas trop cher"	$< g\_adj,$	$(prop\ cost),$	$(not\ expensive) >$

with $(\lambda x.(not\ x)\ (\lambda x.x\ expensive)) \equiv_{\beta} (not\ expensive)$ [7].

4.2.2 Pregroups

Theoritically, the two previous rules are sufficient to segment the utterance into *minimal chunks*. They present disavantage of requiring a great number of definitions in the lexicon. For example, recognition errors are very frequent on the short words: elisions of function words are numerous. So, because of determiner elisions, prepositions must be linked both to nominal groups and to common nouns. The pregroup formalism (Lambek 1999; Buszkowski 2001) is a way to solve the problem. If it is set out that $(nomc) \leq (gn)$ (where *nomc* set for common noun, *gn* for nominal group and *gnp* for prepositional nominal group) and if the type $(gnp)(gn)^l$ is given to a preposition, calculating rules in the pregroups [8] applied to *chunk "de restaurant" (of restaurant)* lead to:

$$((gnp)(gn)^l)\ (nomc) \leq (gnp)\ ((gn)^l(gn)) \leq (gnp)(1) \leq (gnp)$$

These calculating rules are applied to the two first elements of the definition triplets and they are extended in order to take account of the traits allowing to know the syntactic structure of the *chunk*: in the two following rules, types c and m can include respectively occurrences of c_2 and m_2; like in the previous rules, F is an abstraction in the λ-terms meaning.

$$\forall c_1 \leq c_2 \;\; \forall m_1 \leq m_2$$
$$< cc_2^l,\ mm_2^l,\ F > < c_1,\ m_1, s > \rightarrow < c[c_2 \leftarrow c_1],\ m[m_2 \leftarrow m_1],\ (F\ s) >$$

$$\forall c_1 \leq c_2 \;\; \forall m_1 \leq m_2$$
$$< c_1,\ m_1, s > < c_2^r c,\ m_2^r m,\ F > \rightarrow < c[c_2 \leftarrow c_1],\ m[m_2 \leftarrow m_1],\ (F\ s) >$$

So, the first rule is applied to the *chunk "de restaurant"* :

---

[7] LOGUS is implemented in λProlog, a logical programming language whose terms are λ-terms with simple types (Belleannée 1999).

[8] In a pregroup, $b \leq c \Rightarrow ab \leq ac$ and $ba \leq ca$, $a^l a \leq 1$, $aa^r \leq 1$.

$$\begin{array}{ll}
\text{“}de\text{”} & : < (gnp\ (gn\ A\ B)\ (prep\ of))(gn\ A\ B)^l,\ object\ (object)^l,\ \lambda x.x > \\
\text{“}restaurant\text{”} & : < nomc,\ object,\ restaurant > \\
\hline
\text{“}de\ restaurant\text{”} & : < (gnp\ nomc\ (prep\ of)),\ object,\ restaurant >
\end{array}$$

with $(gnp\ (gn\ A\ B)\ (prep\ of))[(gn\ A\ B) \leftarrow nomc] \equiv (gnp\ nomc\ (prep\ of))$.

4.3 Implementation, results and conclusion

4.3.1 Implementation

Implementing the previous rules consists in applying all the possible compositions up to exhaustion. Several segmentations of the utterances are possible. It is rational to consider the best segmentation is the one carrying out the most numerous links. This heuristics leads to preserve only segmentations with the less numerous constituents.

Then, a filter is applied which deletes definitions related to fractional triplets. So, this step provides a means to a first treatment of speech repairs. For example, false starts on nominal groups can be deleted; so, *[l'adresse] [du] [de l'hôtel] (the address of the of the hotel)* is reduced to the two *chunks*: *[l'adresse] [de l'hôtel]*.

Then, a second heuristics is applied; according to this one, the best segmentations are the ones leading to delete the minimum of constituents. So, solutions are called *optimal solutions* if they meet the two **successive** following requirements:

(1) the number of their constituents is minimum before filter application,
(2) the number of their constituents is maximum after filter application.

4.3.2 Ambiguity

Theoritically, several optimal segmentations are possible for an utterance. However, tests made in order to evaluate LOGUS have led to know that, usually, there is only one optimal segmentation. On the other hand, each of the result segments can have several definitions as it is showed in the following tables, from a study on a hundred sentences picked from these tests. Each definition is related to a specific interpretation of the *chunk* in the target language: the refered *chunks* may be seen as ambiguous *chunks*. In view of these results, ambiguity can seem to be weak. However, it is present enough to need to be taken into account. It is important to bear in mind that, in the frame-based methods, the selected segments are supposed to bear no ambiguity.

Number of sentences	Number of *chunks*	Number of ambiguous *chunks*
100	690	48

1 - Number of ambiguous *chunks*

Number of definitions by *chunk*	1	2	3	≥ 3
Number of *chunks*	642	42	6	0

2 - Number of definition by *chunk*

Number of ambiguous *chunks* by sentence	0	1	2	3	4
Number of sentences	59	36	4	0	1

3 - Number of ambiguous sentences

4.3.3 Conclusion

LOGUS has taken part in the "challenge" evaluation campaign held by the French CNRS research agency (Antoine and all 2002). The aim of this evaluation was to provide diagnostic of the assessed systems. The 1200 tests taken by LOGUS have showed the robustness of the system and relevance of the chunking: the number of errors produced by this first step is very low. So, it gives a robust basis for the following analysis. At present, LOGUS participates on MEDIA project of Technolangue group initiated by the French research minister. The project must lead shortly to a dialogue context evaluation.

5 Perspectives: categorial grammars and partial parsing

The *chunking* made in LOGUS is designed for an especially un-normed language. The use of *minimal chunks* restricts application scope of categorial grammars in the studied application. On the other hand, we dare say that the expressive ability of this formalism can be used more widely for partial parsing in other applications, for example for some un-normed written languages. For example, in the absence of any recognition errors, a nominal group can be seen as a one-*chunk* and can include coordinations. Furthermore, always in the absence of recognition errors, it is also possible to envisage that the categorial grammar rules support a first step of *chunks* linkage.

5.1 Chunks *extension*

The extension of the *chunks* goes with the need to link lexical lexemes which leads to two additional difficulties:

- In LOGUS, fractional categories are related to function words and the content words definitions are generally very simple. This feature makes it easier to specify the lexicon for a new domain application, since more complex definitions of the function words can generally be retained.
- Furthermore, when content words must be linked, it is necessary to determine λ-terms in order to obtain the expected semantic translation. These definitions are not always self-evident, as it is showed in the three following examples.

(1) *"de deux ou trois restaurants" (of two or three restaurants) :*

Without constraint of *minimal chunks*, the elements of the phrase *"de deux ou trois restaurants" (of two or three restaurants)* can be linked in a same *chunk*. The numeral adjectives or the common noun must have a fractional category. In the solution given below, this category is given to the word *"restaurant"*:

"de" (of)	$< (gnp\ (gn\ A\ B)\ (prep\ de))(gn\ A\ B)^l,\ (obj.)(obj.)^l,\ \lambda x.x >$
"deux" (two)	$< adj\_num,\ (prop\ nb),\ (int\ 2) >$
"ou" (or)	$< (c^r c)c^l,\ (m^r m)m^l,\ \lambda x(\lambda y.(or\ y\ x)) >$
"trois" (three)	$< adj\_num,\ (prop\ nb),\ (int\ 3) >$
"restaurants"	$< (adj\_pre^r(gn\ nomc\ adj\_pre)),\ (prop\ R)^r(objet),\ \lambda x.(restaurant\ (R\ x)) >$

The definitions lead to:

"deux ou trois"
$< adj\_num,\ (prop\ nb),\ (or\ (int\ 2)\ (int\ 3)) >$
"deux ou trois restaurants"
$< (gn\ nomc\ adj\_num),\ object,\ (restaurant\ (nb\ (or\ (int\ 2)\ (int\ 3)))) >$
"de deux ou trois restaurants"
$< (gnp\ (gn\ nomc\ adj\_num)\ (prep\ de)),$
$object,\ (restaurant\ (nb\ (or\ (int\ 2)\ (int\ 3)))) >$

(2) *"dans deux ou trois jours" (in two or three days time):*

In order to interpret this phrase accurately, giving a specific definition to the word *"jours"* is not enough. It is also necessary to define specific meaning of the preposition *"dans"* in an expression like this:

"dans" : $< (gnp)(gn)^l,\ (prop\ date)(prop\ duration)^l,\ \lambda x.(after\ x) >$
"deux ou trois" : $< adj\_num,\ (prop\ nb),\ (or\ (int\ 2)\ (int\ 3)) >$
"jours": $< (adj\_num)^r(gn),\ (prop\ nb)^r(prop\ duration),\ \lambda x.(nbdays\ x) >$
"deux ou trois jours":
$< gn,\ (prop\ duration),\ (nbdays\ (or\ (int\ 2)\ (int\ 3))) >$

“dans deux ou trois jours”

$< gnp,\ (prop\ date),\ (after\ (nbdays\ (or\ (int\ 2)\ (int\ 3)))) >$

(3) *“dans un deux ou dans un trois étoiles” (in a two or in a three stars):*

This expression is more difficult to parse. Syntactic associations are simple but the problem is to obtain correct semantic translation because the specific and elliptical use of the word *“étoiles”*. In the absence of coordination, the ellipsis *“dans un trois étoiles” (in a three stars)* used for *“dans un hôtel trois étoiles” (in a three star hotel)* is easy to treat. The following definition given to the word *“étoiles”* will suffice:
$< adj\_num^l nomc,\ (prop\ nb)^l(object\ hotel),\ \lambda x.(hotel\ (nbstars\ x)) >$.

For treatment of the phrase a solution can be to give the following definitions to the words *“dans”* and *“un”*:

“dans” : $< ((gnp\ (gn\ nomc\ (det\ A\ B))\ (prep\ in))(nomc^l))(det\ A\ B)^l,$
$(prop\ R)(prop\ R)^l,\ \lambda x.x >$

“un” : $< (det\ indef\ sing)\ adj\_num^l,\ (prop\ R)(prop\ R)^l,\ \lambda x.x >$

So the expression *“dans un deux ou dans un trois”* is a *chunk* defined by:
$< (gnp\ (gn\ nomc\ (det\ indef\ sing))\ (prep\ in))(nomc^l),$
$(prop\ nb),\ (or\ (int\ 2)\ (int\ 3)) >$

This solution is linguistically justified; in French, elliptical phrases like *“un grand” (a great)*, *“pour deux” (for two)* are frequent, especially in a dialogue. Their interpretation during the following steps of the analysis requires mainly the nature of the expressed property to be known. On the other hand, a semantic role like $(object\ R)(object\ R)^l$ doesn't give useful information.

Given this choice, the following definition given to the word *“étoiles”* leads to associate the various elements and to obtain the expected translation.

“étoiles” : $< ((gnp\ \ A\ B)(nomc^l))^r(gnp\ A\ B),$
$(prop\ nb)^r(object\ hotel),\ \lambda x.(hotel\ (nbstars\ x)) >$

These three examples are presented in order to show that widening of the segmentation is possible but that choices must be done to give fractional definitions to lexical lexemes. Involved in an engineering rather than formal approach, these choices have to give a great deal of syntactic and semantic informations in order to continue the analysis.

5.2 Chunks *linkage*

The *chunking* uses the rules of the categorial grammars applied together to syntactic categories and to semantic roles. A more specific definition of the semantic roles would lead to define a first step of composition between *chunks* through application of categorial grammars to the semantic roles. It would be possible to get the following rule:

$$< C,\ R,\ T >, < \_,\ R\backslash\ RR,\ P > \rightarrow < C,\ RR,\ (\lambda x.T\ \ P) >$$

This rule could then link an object with its properties. For example, after *chunking*, the phrase *"deux chambres avec douche ou salle de bain" (two rooms with shower or bath room)* is split into the two following components:

"deux chambres":

<(gn nomc adj_num), (object room), (room (number (int 2)))>

"avec douche ou salle de bain":

<(gnp nomc (prep with)), (object room)\ (object room),
(or (sanit. shower) (sanit. bath_room))>

Like in the *chunk* extension (cf. §5.1), this step implies that a part of the semantic knowledge is transfered into the lexicon - without diappearing from the ontology -. Its main advantage is the simplicity of the rules used here. However, to ignore syntactic category restricts its potential scope to linkages of a very great semantic weight. The double principle of an incremental analysis and of use of the same formalism during the different steps implies these steps differ in the nature of the used rules. A greater widening of the use of categorial grammars would require that this principle - which is not absolute - be questioned.

5.3 Conclusion

To our knowledge, Logus is the first spontaneous speech understanding system where use of categorial grammars is attempted. This experience is conclusive and these formalisms have showed they are very well suited to partial analysis. These two lexicalized formalisms lend themselves to the implementation of bottom-up and incremental analysis. Furthermore, because of the association with λ-calculus, they enable you to build semantic translation. It is also easy to adapt them in order to combine various arguments: in Logus, semantic and syntactic arguments are combined but in the formalism used to define constituents, it is very simple to introduce other arguments, as, for example, prosody.

Categorial grammars and pregroups are a means to make segmentations, more or less wide according to normality of the parsed language, even thought other formalisms are probably needed for the following steps. So, we think partial analysis could become a significant field of study for these two formalisms.

References

[Abney 1991] Abney, S.: Parsing by Chunks. Principle Based Parsing, R. Berwick and S. Abney and C. Tenny, Eds., Kluwer Academic Publishers (1991)

[Aït-Mokhtar 2002] Aït-Mokhtar, S. and Chanod, J-P. and Roux, C.: Robustness beyond Shallowness: Incremental Deep Parsing. Natural Language Engineering, **8**, (2002), 121–144

[Antoine and all 2002] Antoine, J.-Y. and Bousquet-Vernhettes, C. and Goulian, J. and Kurdi, M.-Z. and Rosset, S. and Vigouroux, N. and Villaneau, J.: Predictive and Objective Evaluation of Speech Understanding: the "challenge" evaluation campaign of the I3 speech workgroup of the french CNRS. Proceedings of the LREC 2002, 3^{rd} International Conference on Language Resources and Evaluation, Las Palmas, Spain, (2002)

[Austin 1970] Austin, J.-L.: How to do things with words, Oxford (1962)

[Bar-Hillel 1964] Bar-Hillel Y.: Language and Information, chapter On Categorial and Phrase Structure Grammars, Addison-Wesley, Reading (1964)

[Bear 1992] Bear, J. and Dowding, J. and Shriberg, E.: Integrating multiple knowledge sources for detection and correction of repairs in Human-Computer dialogue, ACL'92 (1992), 56–63

[Belleannée 1999] Belleannée, C. and Brisset, P. and Ridoux, O.: A Pragmatic Reconstruction of λProlog, Journal of Logic Programming, **41** (1999) 67–102

[Buszkowski 2001] Buszkowski, W.: Lambek Grammars Based on Pregroups. Logical Aspects of Computational Linguistics, Springer, Le Croisic, (2001), 95–109

[Lambek 1999] Lambek, J.: Type grammars revisited. Logical Aspects of Comptutational Linguistics, LNAI 1582, Springer, Berlin, (1999), 1–27

[Lamel 2000] Lamel, L. and Rosset, S. and Gauvain, J-L. and Bennacef, S. and Garnier-Rizet, M. and Proust, B.: The LIMSI-ARISE System. Speech Communication, Elsevier science B. V., **31**, (2000) 339–353

[Martinie 2001] Martinie B., Remarques sur la syntaxe des énoncés réparés en français parlé. Recherches sur le français parlé, **16**, (2001), 189–206

[Searle 1970] Searle, J. R.: Speech Acts, Cambridge University Press, (1970)

[Seneff 1992] Seneff S.: Robust Parsing for Spoken Language Systems. Proc. ICASSP (1992), 189–192

[Sowa 1984] Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)

[Vanderveken 2001] Vanderveken, D.: Essays in Speech Act Theory, chapter Universal Grammar and Speech Act Theory, John Benjamin, (2001), D. Vanderveken and Susumu Kubo, 25–62

[Zue 2000] Zue, V. and Seneff, S. and Glass, J. and Polifroni, J. and Pao, C. and Hazen, T.J. and Hetherington L.: JUPITER : Telephone-Based Conversational Interface for Weather Information. IEEE Transactions on Speeh and Audio Processing, **XX, Y** (2000), 100–112