Logical Approach to Natural Language Understanding in a Spoken Dialogue System

Jeanne Villaneau, Jean-Yves Antoine, Olivier Ridoux

Logical Approach to Natural Language Understanding in a Spoken Dialogue System

Jeanne Villaneau[1], Jean-Yves Antoine[1], and Olivier Ridoux[2]

[1] Université de Bretagne-Sud, Valoria, Campus de Tohannic, 56000 Vannes, France
Email: jeanne.villaneau@univ-ubs.fr jean-yves.antoine@univ-ubs.fr
[2] Université de Rennes 1, IRISA, 35042 Rennes cedex, France
Email: olivier.ridoux@irisa.fr

Abstract. We present a logical approach of spoken language understanding for a human-machine dialogue system. The aim of the analysis is to provide a logical formula, or a conceptual graph, by assembling concepts related to a delimited application domain. This flexible structure is gradually built during an incremental parsing, which is meant to combine syntactic and semantic criteria. Then, a contextual understanding step leads to completing this structure. The evaluations of the current system are encouraging. This approach is a preliminary for a logical dialogue that uses the form of the semantic representations.

1 Introduction: Logical approach for spoken language understanding

Most currently operational man-machine systems achieve a precise task, limited to a very constrained domain. Moreover, dialogue is very often machine-directed and gives little flexibility to the users in expressing their queries. For the development of more elaborate spoken dialogue systems, one needs to solve several problems: Spoken Language Understanding (SLU) is one of the most important of them [6].

Most SLU systems use a frame-based approach; because of the simplicity of the envisaged task, it is possible to build semantic frames in order to represent all possible queries. Such approaches give a way to obtain effective and robust parsing: complete linguistic analysis is not needed. Because the frames give the semantic structure of the queries, understanding may be reduced to spoting of keywords or phrases in order to instantiate the different parameters of these patterns. It is not clear whether or not these methods, based on the absence of linguistic ambiguity, are sufficient if the domain becomes less constrained, if interaction between the system and the user is expected or if the system must understand less simple requests. According to Allen and his colleagues, *"they do not capture enough of the subtlety and distinctions that people depend on in using language"* [2]. Other approaches of speech understanding are needed, which have to combine accuracy and robustness.

In this report, we present a logical approach of spoken language understanding and its implementation: the LOGUS[3] system. LOGUS is designed for spontaneous French spoken language understanding in man-computer dialogue; it is relevant to a family of tasks related to a delimited domain: the understanding is not frame-based but a semantic knowledge of

[3] LOGical Understanding System

Petr Sojka, Ivan Kopeček and Karel Pala (Eds.): TSD 2004, LNAI 2448, pp. 1–8, 2004.

the application domain can be used. The test domain is touristic information; it is well delimited yet wide enough to require the use of rather complex structures: for example the representation of a phrase as *"le tarif des chambres doubles et simples au Caumartin ou au Crillon" (the price for double and single rooms in Caumartin or Crillon)* has to represent the dependencies between the objects and the scopes of the coordinations. Besides information tasks, other tasks as renting or reservation must be taken into account; they can lead to some changes in the database.

In the absence of semantic frames, a target language is needed in order to represent the meaning of the utterances. LOGUS uses logical tools chosen for their expressivity and in the perspective of a logical dialogue. They are presented in Section 2. The representation structures are flexible: building them requires a complete linguistic analysis of the utterances, which has to be both robust to agrammaticalities, and precise. This analysis uses logical tools too, both for the un-contextual and contextual understanding. LOGUS parsing is presented in Section 3. The last section of this report (Section 4) presents some results of an evaluation of the system and the perspectives of development.

2 Semantic representation

A SLU system aims at providing results to a dialogue manager: the choice of the target language used for the semantic representations has to take this objective into account. The general intention of the user who interacts with the machine is known or assumed to be known: it is an information query. Nevertheless, intentions expressed in the utterances during the dialogue can be various. For example, for a simple 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.

When a user interacts with a spoken dialogue system, language function is "*to do things"* in the meaning given by Austin and Searle in the speech-act theory [4]. Illocutionary logic of D. Vanderveken takes this speech pragmatics into account: in this formalism, the form of an elementary illocutionary act is $F(P)$ where F is the illocutionary force of the act and P its propositional content [9].

According to this formalism and in the perspective of a logical dialogue management, we have chosen a logic formula as the semantic representation of an utterance. The propositional content of an elementary illocutionary act is a structure, built with the domain objects and their properties: it is called an *object string*. A *language act* contains clues about the intentions of the speaker and plays the role of illocutionary force. The object strings and language acts are represented with concepts and conceptual structures in order to enable the logic formula to be convertible into a conceptual graph [8]. Figure 1 shows an example of semantic representation as both logical formula and conceptual graph. As usual, concepts are in the rectangular boxes and conceptual relations in the rounded boxes. Two single objects are linked with the subordination relation of. The properties "*double"* and "*demain" (tomorrow)* are represented by the conceptual relations *size* and *date* applied to the concepts *double* and *tomorrow*. The language act is represented by its form (*interrogation*) and its content *(possibility)*. In the logical formula, a single object is represented by its label applied to the list of its properties.

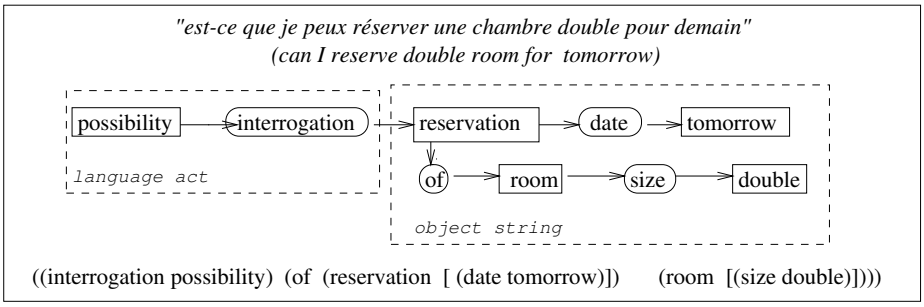

Fig. 1. Example of semantic representation

3 LOGUS parsing

Our spoken French language understanding system works for a domain which remains limited yet much wider than the standard systems, and where linguistic ambiguities are present. The parsing must be precise in order to clear up these ambiguities and to understand exactly what the user wishes to do . It must be robust in order to withstand recognition errors and spoken language features.

3.1 General principles of the parsing

Shallow parsing (as opposed to deep parsing) is frequently used in order to design robust parsing systems. Aït-Mokhtar and his colleagues assert that an incremental methodology is a way to design deeper language parsing while preserving robustness [1]. This assertion concerns text parsing. We argue that it can be extended to spoken language parsing.

In LOGUS, the constituents of the parsed sentence are gradually combined. As they are increasing, their meaning becomes more specific. Several different formalisms are used in succession; they are adapted from standard syntactic formalisms in order to associate syntactic and semantic arguments. Syntactic constraints are gradually relaxed to cope for agrammaticalities.

During the different steps of the parsing, only one formalism is used to represent constituents. It is designed to distinguish syntax and semantics and to preserve genericity of the parsing rules. A constituent can have several definitions; each of them is a triplet $\langle C,\ R,\ T\rangle$ where

- C is a syntactic label, called *syntactic category*: for example, *adjective, (verb 1 present).*
- R is the *semantic role*. It points out semantic function of the constituent: for example, *object, (prop cost)* where *prop* means *property*.
- T is the *semantic translation*. It belongs to the target language.

3.2 Steps of the parsing

The general structure of the LOGUS system is shown in figure 2. Parsing is essentially split into three steps:

– The first step is a segmentation into *chunks* in order to link function words to the nearby content words (cf. §3.3).
– The second step is used in order to compound the *chunks*. It is split into several phases where syntactic contraints are gradually relaxed (cf. §3.5).
– The third step is a contextual understanding step where dialogue context is used in order to achieve interpretation of the utterance (cf. §3.6).

The last two steps use domain ontology which lists the possible dependencies between objects and properties (cf. §3.4).

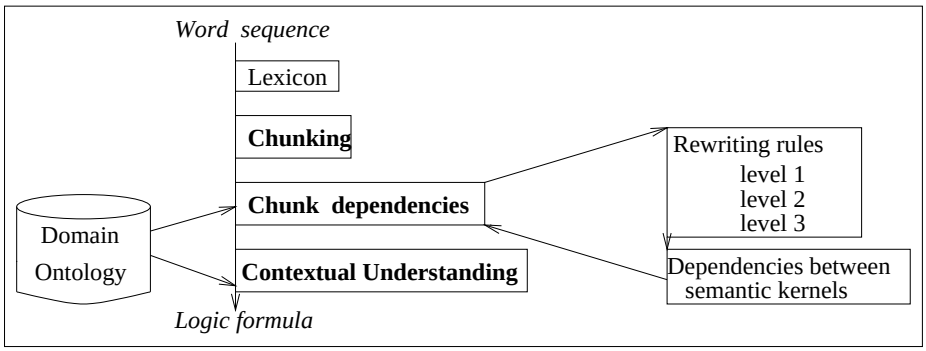

Fig. 2. LOGUS structure

3.3 Chunking

Studies of repairs in French have shown that they preserve minimal syntactic structures: in 70% of the speech repairs of prepositional syntagms, the syntagm is resumed as a whole [7]: *"vers le vers la station" ("at the at the station").* According to these results, *chunking* seems possible to parse spoken language.

Nevertheless, the evaluation of a LOGUS prototype has shown that *chunking* is effective provided that *chunks* are very short. More precisely, errors made at the speech recognition level make it dangerous to link objects or properties after purely syntactic criteria, without checking these links in the ontology. Therefore the *chunks* used in LOGUS, called *minimal chunks*, include only one content word. General principle consists in linking function words to the near content word.

The formalism used for chunk parsing is Categorial Grammars of AB type [5], whose rules are generalized to constituent triplet. Function words have definitions where syntactic category and semantic role are fractional. In these definitions, semantic translation is an abstraction (in the λ-term meaning). The semantic translation of the result triplet is achieved by applying this abstraction to the semantic translation of the un-fractional triplet. Formally, the two following rules are applied:

$$\langle C_A/C_B,\ R_A/R_B,\ F\rangle,\ \langle C_B,\ R_B,\ S_B\rangle \ \rightarrow\ \langle C_A,\ R_A,\ (F\ S_B)\rangle$$
$$\langle C_B,\ R_B,\ S_B\rangle,\ \langle C_B\backslash C_A,\ R_B\backslash R_A,\ F\rangle \ \rightarrow\ \langle C_A,\ R_A,\ (F\ S_B)\rangle$$

The first rule is applied twice in the following example where the phrase *"pas trop cher" ("not too expensive")* is parsed (s_adj=adjectival syntagm).

"pas"	⟨	*(s_adj / adjective),*	*(prop R) / (prop R),*	$(\lambda x.(not\ x))$ ⟩
"trop"	⟨	*(adjective/ adjective),*	*(prop R) / (prop R),*	$\lambda x.x$ ⟩
"cher"	⟨	*adjective,*	*(prop price),*	*cher* ⟩
"pas trop cher"	⟨	*s_adj,*	*(prop price),*	*(not expensive)* ⟩

In the implementation, the rules are applied as far as it is possible. Only solutions with minimal number of constituents are retained. The result is a sequence of large constituent triplets. Then, constituents with fractional categories are removed. These suppressions are a first way for dealing with repairs: they eliminate beginnings of uncompleted syntagms.

3.4 Domain ontology

The limited scope of the domain application is captured in a *domain ontology*. It aims at specifying how objects and properties can be compound. Despite the use of this domain dependent ontology, the system is expected to be generic. To achieve this, the ontology is built with generic predicates, whose domain objects and domain properties are the arguments.

- The possibility of a subordination between two objects is defined by the *is_sub_object* predicate. Its arguments are two object labels. For example, the relation *is_sub_object(room, hotel)* indicates that it is possible to build the elementary conceptual graph: room → (of) → hotel
- The predicate *is_property_of* has three arguments: a property label, a property content and an object label. It is used in order to define how properties can be linked to objects. For example, *is_properties_of(date, _, reservation)* indicates that the properties of *date* label can be linked with the objects of *reservation* label.
- The first two predicates only convey semantic relations. Unfortunately, syntactic constraints are sometimes necessary. For example, in the phrase *"aller de l'hôtel Caumartin au..." (to go from Caumartin hotel to...)* the preposition is essential to know that *Caumartin hotel* is a point of departure. The predicate *is_dependent_on* is used to define these dependencies between objects. It uses the syntactic category of the constituent: for example, the relation *is_dependent_on(C, O, from, to_go)* is effective if C contains the *from* preposition.

3.5 Chunk dependencies

Chunk dependencies are analyzed in two phases. In the first phase, rewriting rules are used (see Fig. 2). They are expressed in terms of the first two components of the constituent triplets, and from the generic ontology predicates. For example, the following rule leads to a subordination between two objects:

$$[\langle C1,\ object,\ O1\rangle, \langle C2,\ object,\ O2\rangle] \left\{ \begin{array}{l} -O1 \text{ simple object of label } Et1 \\ -O2 \text{ object string of label } Et2 \\ -is\_sub\_object\_of(Et1,\ Et2) \end{array} \right\} \rightarrow [\langle C,\ object,\ (of\ O1\ O2)\rangle]$$

They are three levels of rules, according to decreasing emphasis on syntactic constraints:

- The first level reconstructs to standard chunks, where it is possible:
  [ŕ l'hôtel] [Caumartin] → *[ŕ l'hôtel Caumartin] (to Caumartin hotel)*
  [deux] [ou] [trois] [étoiles] → *[deux ou trois étoiles] (two or three stars)*
- The second level implements compositions where syntactic and semantic standard constraints are respected (word order, prepositions, etc.).
- The third level removes unknown words and relaxes syntactic constraints.

Levels are exploited in a cascade parsing: all rules of a given level are applied up to saturation before the rules of the next level are applied. Figure 3 shows how dependencies are gradually built for the parsing of an utterance: rectangular dotted boxes frame the *chunks*. Links of the different levels are represented with different dotted arrows.

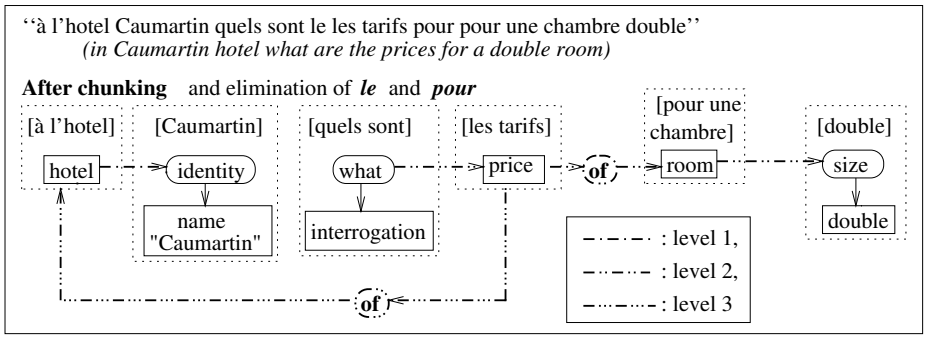

Fig. 3. Cascade parsing

The aim of the second phase is to find semantically most significant constituents, and to link other constituents to them under the control of the ontology. For example, in the utterance *"quels sont les horaires doubles d'ouverture du Louvre" ("what are the timetables double*[4] *of opening of Louvre")*, the word *"doubles"* is due to a recognition error relating to a hesitation. The application of rewriting rules provides three constituents: *[quels sont les horaires] [doubles] [d'ouverture du Louvre]*. Ontology revels that the first and the third can be semantically linked. The second constituent being of little syntactic and semantic weight, one can neglect it.

3.6 Contextual Understanding

Contextual understanding examines dependencies between language acts in an utterance with two purposes:

- The first purpose is resolution of references. In the utterance *"j'ai réservé au Crillon comment je peux faire pour y aller d'ici" ("I have booked in Crillon how can I go to there from here"), "y" (to there)* refers to Crillon hotel and *"ici" (here)* to the contextual place. In the pragmatic dialogues which are processed by the system, this resolution is

[4] In French, the adjective *double* is located after the common noun to which it is linked.

generally simple, if semantic criteria are used: these resolutions are mainly based on the knowledge found in the ontology.

– The second purpose is resolution of some ellipses. For example, *"je voudrais réserver au Crillon quel est le prix pour une chambre double" ("I would like to book in Crillon what is the price for a double room")*, the room is related to hotel Crillon. The system looks for the contextual objects to which sentence objects can be subordinated: again, these possibilities of dependencies are infered by the ontology.

The same principles are implemented for the dialogue contextual understanding. Language acts and objects of the previous utterances are stored and used to solve references and ellipses with the support of the ontology. For example, with the two consecutive utterances:

1. *"quel est le tarif pour une chambre double au Crillon" ("what is the price for a double room in Crillon")*
2. *"et le prix pour une chambre simple" ("and the price for a single room")*

the second sentence contains two ellipses: the request (*"quel est"*) and an object (*"Crillon"*). These links are restored under ontology control.

4 Results and prospects

Performances of speech understanding systems are difficult to evaluate: results are dependent on the domain, on the task, and on the chosen semantic representations. LOGUS has taken part in the challenge-based evaluation campaign held by the GDR-I3 consortium of French CNRS research agency. The issue of this evaluation was to provide diagnosis of the assessed systems despite their disparities [3]; it was not a direct comparison between these systems, which dealt with various domains. In this evaluation every team of the consortium challenges the other teams by providing test data. LOGUS was assessed on a set of 1200 tests which have shown the efficiency of the approach: the robustness of the parsing is satisfactory (error rate lower than 10%) and the system is able to build precise semantic representations.

LOGUS achieves the goal of genericity up to a reasonable extent. To change the application domain amounts to giving definitions (i.e., triplets) of new words, and to using a new domain ontology.

Despite these encouraging results, many developments and studies are necessary to complete this approach. At present, LOGUS is involved in the MEDIA project[5], which will shortly lead to a dialogue contextual evaluation. Otherwise, in order to appreciate relevance of semantic representation, it is necessary to develop dialogue managing where this representation is used. We are envisaging to link this dialogue managing with database of LIS type in order to implement complete and coherent logic approach.

References

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

[5] MEDIA project is a part of the Technolangue project, initiated by three French ministers.

2. Allen, J. et al.: Towards Conversational Human-Computer Interaction. AI Magazine **22-4** (2001), 27–37.
3. Antoine, J.-Y. et al.: Predictive and Objective Evaluation of Speech Understanding: the “Challenge XI” evaluation campaign of the I3 speech workgroup of the French CNRS, Proceedings of the LREC 2002, 3rd International Conference on Language Resources and Evaluation, Las Palmas, Spain, (2002).
4. Austin, J.-L.: How to do things with words. Oxford (1962).
5. Bar-Hillel, Y.: Chapter: On Categorial and Phrase Structure Grammars Language and Information, Addison-Wesley, Reading (1964).
6. Glass, J. R.: Challenges for Spoken Dialogue Systems. Proceedings IEEE ASRU Workshop (1999), Keystone, Colorado, USA.
7. Martinie, B.: Remarques sur la syntaxe des énoncés réparés en Français parlé. Recherches sur le Français parlé, **16**, (2001) 189–206.
8. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, Reading (1984).
9. Vanderveken, D.: Chapter: Universal Grammar and Speech Act Theory Essays in Speech Act Theory, John Benjamin, D. Vanderveken and Susumu Kubo, (2001), 25–62.