

SEMANTIC ANALYSIS IN DIALOGUE INTERFACES

ADRIAN ONET, DOINA TĂTAR

ABSTRACT. One crucial issue for the NL interfaces is the use of an "intermediate meaning representation formalism" which will support the semantic and pragmatic reasoning processes of the system. The paper presents a syntactic-semantic analyzer based on the approach of lambda-calculus, realised by the first author, as a kind of syntax-driven, context independent and inference free approach. The first level of this application contains the semantic engine (written in SWI-Prolog); the second one contains an interface with the user (written in Delphi); the extra level is for the graphical representation of the parse tree (written in Visual Prolog).

1. DIALOGUE INTERFACES

A fundamental goal of artificial intelligence is the manipulation of natural languages (NL's) using the tools of computing science. The main challenges raised by NL processing arise at many levels: conceptual model, semantic theories, parsing theories, user modeling. The NL phenomenon has some important characteristics that must be considered when one implements an NLP system [15]:

- Lack of an explicit definition;
- Presence of incomplete and ill structured sentences, without preventing the understanding;
- Influence of the context;
- Ambiguities.

These few characteristics show that NLP requires techniques different from the traditional techniques. Several scientific disciplines have made natural language an object of study: artificial intelligence, linguistics, philosophy, logic, psychology. All these attempt to answer the question of "automatic NL understanding". The most used criterion now is the reasoning process operating on some internal representation of the meaning of the NL input.

The first major success for natural language processing (NLP) was in the area of database access. One of the first such interfaces was Fernando Pereira's CHAT system

2000 *Mathematics Subject Classification*. 68U35.

1998 *CR Categories and Descriptors*. I.2.1 [Computing Methodologies]: Artificial Intelligence – Applications and Expert Systems.

(1983) about a geographical database. Over the last decade, some commercial systems have built up large grammars and lexicons to handle a wide variety of inputs."The main challenge for current systems is to follow the context of on interaction" ([10]).

One crucial issue for the NL interfaces is the use of an "intermediate meaning representation formalism" which will support the semantic and pragmatic reasoning processes of the system. Such of representation is called "intermediate logical form" and it is the principal point through which results coming from the field of logic can be used in a NL processing (NLP) system .

The semantics of the phrases expressed in a natural language has two aspects: semantics and pragmatics. Semantics refer to those aspects of the meaning that are not influenced by the context, and the pragmatics is concerned with the context and the intention of the speaker. Almost every approach for the semantic interpretation of a phrase is made with the principle of compositionality :*the meaning of a phrase is a function of the meanings of its parts* .

The dialogue-based application include [1]:

- question-answering systems, where NL is used to query a database;
- automated customer service;
- tutoring systems;
- spoken language control of a machine;
- general cooperative problem-solving systems.

A dialog interface does have to process sequences of sentences exchanged between a user and an application system. Each of these sentences has to be precisely understood. The discourse domain of one interface is usually restricted, and thus easier to model from a semantic point of view. From a historical perspective, can be distinguished three generations of NL interfaces [14]:

- The "direct translation systems", performing a direct translation of the NL input into an output string, suitable for the purposes of the application. The parser of such a system does not make use of a general meaning representation formalism. These systems are not portable and is difficult to implement in them the semantic inferences.

- The second generation of NL interfaces separates the understanding process into two steps: in a first step an analyzer will process the NL input and produce a representation of its meaning in an intermediate meaning representation formalism, usually an intermediate logical form (ILF). In a second step, an interpreter will study this representation and will find out related actions, accordingly with the application. Both analysis and interpretation are based on an explicit model

of the discourse domain, as a knowledge base defining the ideas referred, providing semantic and pragmatic information and performing the logical inferences necessary for understanding.

- The third generation of NL interfaces includes, besides the model of discourse domain, an explicit model of user with "static" information, such as the level of competence possessed by a specific user, and "dynamic" information expressing the knowledge and beliefs of the user and the evolution of these knowledge and beliefs within the dialogue. This kind of information can be used to improve the resolution of ambiguities, the processing of incomplete sentences and the generation of cooperative responses.

The study of intermediate meaning representation (IMR) formalism has been the subject of large disputes. The question was of deciding whether IMR should be "logical" or not (based on frames, semantic networks, conceptual dependencies, etc) [13]. Is it largely accepted that an IMR formalism must combine different kinds of elements, all of which are necessary for the interpretation process [15]:

- Logical structure;
- Conceptual content: the variables and constants of the logical notation appear as instances of a class system that provides a conceptual model of the discourse domain. This class structure can be organized hierarchically as a lattice and forms the skeleton of the knowledge base used in NL interface;
- Speech act indication representing the expected impact that the speaker tries to have on his interlocutor by uttering a proposition, depending on the nature of this utterance: request, order, information, etc. This expected impact can be modeled in terms of "wants", "knowledge" and "beliefs" of the interlocutor. The primitives expressing this levels can be logically axiomatized and support a reasoning process improving the behavior of an NL interface;
- Pragmatic annotations about determination of logical quantifiers.

The phase of interpretation of an ILF, after his production by the parser, is accomplished in some well defined steps [15]. These steps includes a set of processes as: resolution of anaphoric references, resolution of scoping ambiguities and other types of ambiguities which could not be solved in the parsing phase. Also, NL interface that process more than one isolated sentence needs a dialogue manager and the possibility to control interpretation, for example detecting wrong presupposition.

2. SEMANTIC ANALYSIS BY LAMBDA-CALCULUS

Semantic analysis (SA) is the process whereby semantic representations are composed and associated with a linguistic input. The sources of knowledge that are used are: the meanings of words, the meanings associated with the grammatical

structure and the knowledge about the context in which the *discourse* occurs (*semantics of the discourse*).

One approach of SA is by lambda-calculus and it is a kind of syntax-driven SA, context independent and inference free. Such approach is sufficient to produce useful results. Others two approaches are semantic grammars and information extraction [6]. The lambda-calculus SA is based on the principle of compositionality which assert that the meaning of a sentence can be composed from the meanings of its parts. The input of a semantic analyzer is an output of a syntactic analyzer, that means a parse tree or a feature structure, etc. We will assume that it is a parse tree.

In lambda-calculus approach of SA every context free grammar rule is augmented by a semantic rule which specify how to compute the meaning representation of a construction from the meanings of its constituent parts [6]. An augmented rule is :

$$A \longrightarrow \alpha_1 \alpha_2 \cdots \alpha_n \{A.sem = f(\alpha_j.sem \cdots \alpha_k.sem)\}, 1 \leq j \leq k \leq n$$

The denotation $A.sem = f(\alpha_j.sem, \cdots, \alpha_k.sem)$ means that the semantics of A , $A.sem$, will be obtained as a function f on the $\alpha_j.sem, \cdots, \alpha_k.sem$.

Let us consider an example generated by a small subset of rules from ATIS grammar [6]: *Continental serves meat*.

The small subset of ATIS rules is:

$$\begin{aligned} S &\longrightarrow NP VP \\ VP &\longrightarrow Verb NP \\ NP &\longrightarrow ProperNoun \\ NP &\longrightarrow MassNoun \\ Verb &\longrightarrow serves \\ ProperNoun &\longrightarrow Continental \\ MassNoun &\longrightarrow meat \end{aligned}$$

The augmented rules are:

$$\begin{aligned} NP &\longrightarrow ProperNoun \{NP.sem = ProperNoun.sem\} \\ NP &\longrightarrow MassNoun \{NP.sem = MassNoun.sem\} \\ ProperNoun &\longrightarrow Continental \{ProperNoun.sem = Continental\} \\ MassNoun &\longrightarrow meat \{MassNoun.sem = meat\} \end{aligned}$$

These rules assert that the semantics of NP's are the same as the semantics of their individual components. In general will be the case that for non-branching

grammar rules, the semantics associated with the child will be copied unchanged to the parent.

To come up with the semantics for VP's, we will use a notational extension to first order predicate calculus (FOPC) , lambda-calculus, (Church , 1940) that provides the kind of formal parameter that we need.

The λ -expression

$$\lambda xP(x)$$

must be understand as a formula (with $P(x)$ a formula from FOPC), where the free variable x is bound to the specific terms in FOPC. The process of bounding of x with a specific term in FOPC is a λ - reduction and is illustrate by the equality:

$$\lambda xP(x)(A) = P(A)$$

The variables denoted by λ can be in a arbitrary number and their order is the same with the order of their binding to the terms.

With λ notation the augmented rule for *Verb* is:

$$\begin{aligned} Verb \longrightarrow serves \{ & Verb.sem = \lambda x\lambda y\exists eIS - A(e, Serving) \\ & \wedge Server(e, y) \wedge Served(e, x) \} \end{aligned}$$

and for *VP* is:

$$VP \longrightarrow Verb NP \{ VP.sem = Verb.sem(NP.sem) \}$$

The calculus for $VP.sem = Verb.sem(NP.sem)$ is :

$$\begin{aligned} \lambda x\lambda y\exists eIS - A(e, Serving) \wedge Server(e, y) \wedge Served(e, x)(NP.sem) = \\ \lambda y\exists eIS - A(e, Serving) \wedge Server(e, y) \wedge Served(e, Meat). \end{aligned}$$

So, $VP.sem = \lambda y\exists eIS - A(e, Serving) \wedge Server(e, y) \wedge Served(e, Meat)$.

With λ notation the augmented rule for *S* is:

$$S \longrightarrow NP VP \{ S.sem = VP.sem(NP.sem) \}$$

The calculus for $S.sem$ is:

$$\begin{aligned} S.sem = VP.sem(NP.sem) &= \lambda y\exists eIS - A(e, Serving) \wedge \\ &\wedge Server(e, y) \wedge Served(e, Meat)(NP.sem) \\ &= \lambda y\exists eIS - A(e, Serving) \wedge Server(e, y) \wedge Served(e, Meat)(Continental) \\ &= \exists eIS - A(e, Serving) \wedge Server(e, Continental) \wedge Served(e, Meat). \end{aligned}$$

In the applications is used another new notation that facilitates the compositional creation of the desired semantics: *complex-term*. Formally, a complex-term is an expression with the following three-part structure: $\langle Quantifier Variable Body \rangle$ The formulas which use complex-terms usually refereed as *quasi-logical forms*.

To convert a quasi-logical form in a FOPC formula we will use the following schema of rewriting any predicate having a complex-term argument:

$$P(\langle \text{Quantifier Variable Body} \rangle) \wedge U$$

$$\rightarrow \text{Quantifier Variable (Body Connective } P(\text{Variabila}) \wedge U).$$

where *Connective* is \wedge for \exists and \rightarrow for \forall .

Let us consider the sentence: *A restaurant serves meat.*

The needed augmented rules are:

$$\text{Det} \rightarrow a \{ \text{Det.sem} = \exists \}$$

$$\text{Nominal} \rightarrow \text{Noun} \{ \text{Nominal.sem} = \lambda x IS - A(x, \text{Noun.sem}) \}$$

$$\text{Noun} \rightarrow \text{restaurant} \{ \text{Noun.sem} = \text{restaurant} \}$$

$$\text{NP} \rightarrow \text{Det Nominal} \{ \text{NP.sem} = \langle \text{Det.sem } x \text{ Nominal.sem}(x) \rangle \}.$$

The bottom-up calculus is:

$$\text{Nominal.sem} = \lambda x IS - A(x, \text{Noun.sem}) = \lambda x IS - A(x, \text{Restaurant})$$

$$\text{S.sem} = \text{VP.sem}(\text{NP.sem}) = (\text{Verb.sem}(\text{NP.sem}))(\text{NP.sem}) =$$

Using *VP.sem* as above we obtain:

$$(\lambda y)(\exists e)(IS - A(e, \text{Serving}) \wedge \text{Server}(e, y) \wedge \text{Served}(e, \text{Meat}))(\text{NP.sem}) =$$

$$(\lambda y)(\exists e)(IS - A(e, \text{Serving}) \wedge \text{Server}(e, y) \wedge \text{Served}(e, \text{Meat}))$$

$$(\langle \text{Det.sem } z (\lambda x) IS - A(x, \text{Restaurant})(z) \rangle)$$

$$(\exists e)(IS - A(e, \text{Serving}) \wedge \text{Server}(e, \langle \text{Det.sem } z IS - A(z, \text{Restaurant}) \rangle)) \wedge$$

$$\wedge \text{Served}(e, \text{Meat}))$$

$$\exists e (IS - A(e, \text{Serving}) \wedge (\exists z)(IS - A(z, \text{Restaurant}) \wedge \text{Server}(e, z)) \wedge$$

$$\wedge \text{Served}(e, \text{Meat}))$$

$$(\exists e)(\exists z)(IS - A(e, \text{Serving}) \wedge IS - A(z, \text{Restaurant}) \wedge \text{Server}(e, z) \wedge$$

$$\wedge \text{Served}(e, \text{Meat})).$$

Let us observe that a sentence as: *Every restaurant has a menu* has two semantic representation, one which corresponds to the common-sense interpretation (*every restaurant has its own menu*), but also the interpretation which state that *there is one menu that all restaurants share*.

The two interpretation are obtained processing the two complex-term in the following formula in a different order:

$$(\exists e)(IS - A(e, \text{Having}) \wedge \text{Haver}(e, \langle IS - A(x, \text{Restaurant}) \rangle))$$

$$\wedge \text{Had}(e, \langle (\exists y) IS - A(y, \text{Menu}) \rangle)$$

If the first complex-term is processed first, then the obtained formula is:

$$(\exists e)(\forall x)(IS - A(e, Having) \wedge IS - A(x, Restaurant) \longrightarrow Haver(e, x) \\ \wedge (\exists y)(IS - A(y, Menu) \wedge Had(e, y)))$$

If the second complex-term is processed first, then the different formula is:

$$(\exists e)(\exists y)(IS - A(e, Having) \wedge IS - A(y, Menu) \\ \wedge Had(e, y) \wedge (\forall x)(IS - A(x, Restaurant) \longrightarrow Haver(e, x))).$$

The same results will be obtained for the example in the next section.

3. CONTEXT INDEPENDENT SENTENCES MAPPING IN LOGICAL FORM. THE SYNTACTIC-SEMANTIC ANALYZER

Since the very beginning of computer science the natural language represented an important preoccupation for the specialists. The applications in this domain want to resolve two essential issues: the voice recognition (if the user speaks) and text processing (its meaning).

We provide in this paper an application which begins with the semantic representation idea of the context independent sentences in the natural language like expressions in extended first order predicate calculus. First of all we must specify what we mean by the extended first order predicate calculus. Starting with the FOPC we provide a new set of quantifiers, among the existential and universal ones, necessary for the representation of the quantitative sentences semantic. By using this quantifiers we will represent a quantitative sentence semantic like *Most people laugh* as

$$\exists_N X.(people(X) \wedge laugh(X) \wedge most(N)),$$

where \exists_N belongs to the new set of quantifiers.

This FOPC extension will be noted by FOPC/QS (first order predicate calculus for quantitative sentences). For further details see [9].

Back to our application, this will have as entry a natural language sentence introduced from the standard input from which it will result the FOPC/QS of this sentence and a graphical representation of its parse tree. It is very difficult to compare the natural language functionality and the computer systems operation. Problems appear when we deal with semantic ambiguities resolved by the human mind through context and convention. We have tried to eliminate part of these ambiguities introduced by the domain of quantifiers and of operators by the underspecified method. Thus for *Every boy loves a dog* the semantic representations will be like in figure 1:

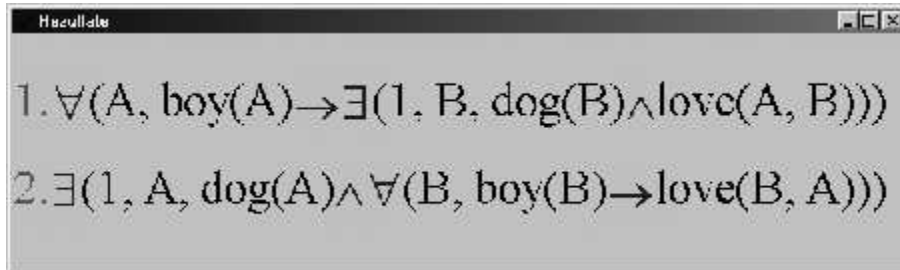


Figure 1

The ambiguities given by the multiple sense of the words will be considered in a future upgrade of the application, which could use the semantic network representation of the Lexis. We must specify that the sentences recognized by the application have to be introduced by an existent grammar. In other words, the user can not modify in any way the existent grammatical rules, but the application could be improved by allowing the user to construct the grammar he needs. This application allows the Lexis entries actualization by an interactive interface. The user could test, after resolving the problems which permit the grammar modifications too, the application in every natural language which describes that grammar. Thus, for every natural language will exists a file which contains its grammar, a file with its lexical entries and also a file which will contain the mapping of every atom structures of its sentences into the semantic representation. For every given sentence the application also presents the advantage of the parse tree graphical representation. Such an example is given as follows: *Every boy loves a dog.* (See figure 2)

We must also say that in the present the application doesn't resolve yet totally the parse of the sentence, more precisely, the gender, person and number agreement. This situation could be improved by modifying the grammatical rules by adding new arguments which represent these agreements. One advantage is that the application can help to design new applications, such as the natural language for querying knowledge bases, natural language conversation. For example, we can create an algorithm which will map every natural language sentence in an equivalent SQL statement in the first type applications. Concerning the structure of this application, it is built on two levels, plus an extra level for the parse tree representation. The first level contains the semantic engine (written in SWI-Prolog); the second one contains an interface with the user (written in Delphi); the extra level is for the graphical representation of the parse tree (written in Visual Prolog). The communication among these levels is done by the use of the Windows

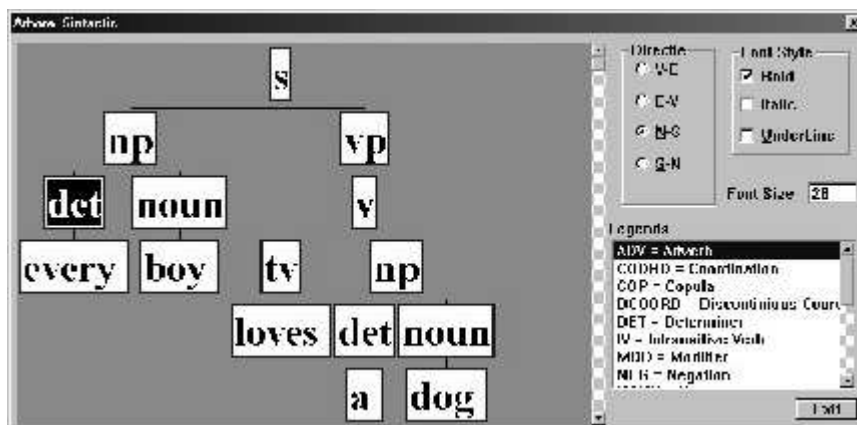


Figure 2

Operating systems specific DDE (dynamic data exchange), we can also use for these communication more evolved techniques such as COM/DCOM.

By its specific, our application construction is based on more programming languages mixture; it also succeeds in tacking advantages on these programming languages characteristics. We believe that this technique can be the starting point for resolving some natural language semantic problems.

REFERENCES

- [1] J.Allen : " Natural language understanding", Benjamin/Cummings Publ. , 2nd ed., 1995.
- [2] C.Beardon, D.Lumsden, G.Holmes: " NL and computational linguistics", Ellis Howood Series, 1991.
- [3] P.Flach: "Simply logical.Intelligent reasoning by example", John Wiley and Sons, 1994.
- [4] A. Flycht-Eriksson: "A domain knowledge manager for dialogue systems", Proceedings of ECAI2000, pp 431-435.
- [5] A.Gaal , G.Lapalme, P.Saint-Dizier: "Prolog for NLP", 1991.
- [6] D.Jurafsky, J.H.Martin: "Speech and language processing", Prentice Hall, 2000.
- [7] R.Kasper, P.Davis, C.Roberts: "An integrated approach to reference and presupposition resolution", The 37th Annual Meeting of ACL, june 1999.
- [8] I.A. Leția and all : " Multi-agent systems", Casa cărții de știinta, Cluj, 1999.
- [9] A. Onet: " Semantic representation of the quantitative natural language sentences" to appear in Studia Universitatis "Babes-Bolyai", seria Informatica.
- [10] S.J. Russell, P.Norvig: "Artificial intelligence.A modern approach",
- [11] I. A. Sag, T. Wasow: "Syntactic Theory:A formal introduction " 1997, <http://ling.ohio-state.edu/HPSG>
- [12] D. Tătar: "Unification Grammars in Natural Language Processing" , in "Recent topics in mathematical and computational linguistic", ed. C. Martin-Vide, G. Paun, Editura Academiei, 2000, pg 289-300.

- [13] A. Thayse (editor): "From standard logic to logic programming", John Wiley and Sons, 1988.
- [14] A. Thayse (editor): "From modal logic to deductive databases", John Wiley and Sons, 1989.
- [15] A. Thayse (editor): "From natural language processing to logic for expert systems", John Wiley and Sons, 1990.

DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF MATHEMATICS AND COMPUTER SCIENCE,
"BABEȘ-BOLYAI" UNIVERSITY, CLUJ-NAPOCA, ROMANIA
E-mail address: `adrian|dtatar@cs.ubbcluj.ro`