Acta Cybernetica 12 (1996) 461-472.

Natural Language Understanding: a New Challenge for Grammar Systems

Carlos MARTÍN-VIDE *

Abstract

We show the basic architecture of a natural language understanding system. Given the well- known difficulties other simple grammar formalisms find when attempting to model such an architecture, as well as the plausibility of the modular hypothesis, we advocate the suitability of complex and modular constructs like grammar systems for giving account of human language.

"Sentence processing is most plausibly modeled as a fully interactive parallel process: each word, as it is heard in the context of normal discourse, is immediately entered into the processing system at all levels of description, and is simultaneously analysed at all these levels in the light of whatever information is available at each level at that point in the processing of the sentence".

(W.D. Marslen-Wilson (1975), "Sentence perception as an interactive parallel process", *Science*, 189: 226-228)

1 Postulates

Let us begin stating some postulates in order to contextualize our paper:

1. There exists a certain undesirable gap between the communities of linguists and computer scientists, more specifically between the communities of computational linguists and formal language theoreticians. Often, linguists ignore all that is strictly beyond/outside the Chomsky hierarchy, and computer scientists don't know precisely the kind of problems linguists are interested in.

2. Formal language theoreticians show an understandable obsession to design mechanisms able to generate recursively enumerable languages. However, no natural language is so large as recursively enumerable. Linguists need formal tools endowed with a very rich internal structure, rather than with an impressive generative power.

*Research Group in Mathematical Linguistics and Language Engineering (GRLMC), Rovira i Virgili University, Pl. Imperial Tàrraco, 1, 43005 Tarragona, Spain, E-mail: cmv@astor.urv.es

3. For theories which try to become rich in applications (and we guess this is the case with grammar systems), empirical well-foundedness is as important as completeness.

4. In an initial step, one of the most essential features of a scientific theory is its metaphorical content, as different from its technical content. Grammar systems seem quite rich in this respect, and quite flexible too.

5. Grammar systems theory needs to assume and face all the complexities of natural language if it wants to be accepted as a good candidate for the solution of natural language processing problems.

6. Linguists are not as much interested in generative capacity aspects of language-theoretical models as in other basic matters like descriptive adequacy, expressiveness, naturality or computational easiness.

We are going to offer an introductory overview of natural language understanding area for non-specialists, which perhaps will help somebody to bring his/her research closer to natural language as it is regarded by current theoretical linguistics. We'll show a picture of natural language from a computational viewpoint. If one wishes the own work will become relevant for linguists, I'm sure one will share the opinion that linguists have something to say about.

2 Language and the computer

It is generally accepted that computers serve not only to process numbers but language too. Even it is a matter of public concern the idea of a machine that could communicate with people in their own language to take commands or to answer questions. In fact, many linguistic tasks, such as translation, improve if performed by a machine with a knowledge of natural language.

We are going to survey the problem of giving a computer comprehension of language. The focus will be on tasks that involve language carrying meaning, rather than those, such as speech processing, that involve only the superficial form of text without regard to its content.

Research in computational linguistics is generally taken as a branch of artificial intelligence, that part of computer science concerned with the computational simulation of intelligent human behaviour (which surely includes language understanding). Most of the natural language research in artificial intelligence has been directed implicitly or explicitly to the problem of computer understanding of language. The converse of language understanding is language production or generation. For the computer, to produce text has proved to be an even harder problem than comprehension.

Although linguists agree that human language is essentially oral, natural language understanding deals almost exclusively with a simple form of language: written text. The additional problems that arise with spoken input and output have been traditionally taken to be primarily matters of physics and engineering.

3 Understanding

3.1 What is understanding

A preliminary question needs to be posed: what means to say that a computer understands? As one believes that understanding is a somewhat subjective state that admits of degrees, it would not seem appropriate to attribute understanding to a machine. We'll agree, however, with the idea that behaviour is the key fact: if the machine always responds to sentences just as a human would in the same situation, then it can meaningfully be said that it understands the sentences.

3.2 The illusion of understanding

Real understanding is hard to achieve in practice. It is not difficult for a computer program to give us the illusion that it understands. One of the most famous examples is ELIZA system, built by Joseph Weizenbaum on 1966, which was not taken as a serious model of understanding, because it used simple scripts and tricks for the computer to keep up a conversation. It's doctor script simulated a psychotherapist. It looked out for certain key words and sentence forms, and answered with one of a few predetermined phrases for each. If the user of the program typed:

(1) I am depressed,

it might answer:

(2) I'm sorry to hear that you are depressed.

If the user's sentence included the word *mother*, the computer might say:

(3) Tell me more about your family.

If the user's sentence matched nothing at all in the script, it would either respond:

(4) What else does that bring to mind?, or

(5) Earlier, you mentioned your mother.

The illusion of understanding cannot be sustained for a long time. The computer only encourages the user to continue, but it understands nothing.

3.3 Levels of understanding

Four levels of understanding can be identified:

a) The most superficial level is that involved in message passing. If a computer is asked to:

   (6) Tell George that I'll meet him next Monday in Salgótarján,

   it needs not understand the message itself, or where Salgótarján is and so on, in order to be able to pass on the message; but it does need to determine that *him* here refers to George.

b) The second level is almost literal understanding within a very limited domain of discourse. This level is a characteristic feature of many natural language systems of the early 1990s, such as interfaces to databases.

c) The third level might be called complete understanding: a full apprehension of all aspects and nuances of the sentence. It allows to read texts and integrate the knowledge gained from them with its previous knowledge from other linguistic sources. This depth of understanding, for instance, seems necessary for unassisted machine translation.

d) The fourth and deepest level is emotional understanding, the level at which people may understand poetry. Today computers are far from this sophisticated level of comprehension.

3.4 Why language understanding is difficult for computers

Language understanding is difficult for computers because both language and the world itself to which language refers are extremely complex, much more complex than expected. But native speakers' facility with their own language is so great and early in their lives that it is hard to see why it is difficult to design computer programs to perform the same task. We only notice the difficulty of language in the special situation of learning a second language, and the problems encountered there –learning things like vocabulary, morphology, conjugations, genders, and irregularities– become memorization tasks which seem to be straightforward to computerize. But these tasks are not as simple as they seem at first sight. The syntax and morphology of natural languages are objects of high complexity. Words and idioms may convey complicated meanings. Native speakers may make quite subtle distinctions at any level of the structure of language. This big body of knowledge is the main topic of theoretical linguistics, and the progress of computational processing of language has been, is and will be closely linked to it.

Complexities of language are compounded by other minor problems that are easy to handle for humans but can be extremely difficult for computers. It is the case of ambiguity. Ambiguity appears at several levels of language:

a) at the lexical level: few dictionary entries list just one meaning for a word,

b) at the syntactic level: most sentences admit more than one parse tree,

c) at the pragmatic level: most sentences allow more than one analysis of the pragmatic role they play in the context of discourse where they are being uttered.

However, in spite of such potential multiplicity of choices, just a single interpretation of the sentence is intended by the speaker: it is the task of the listener (and of the computer) to recover it in order to achieve full understanding.

3.5 Knowledge of the world

The difficulty of language understanding is also a reflection of the complexity of the world, for one cannot understand language without becoming involved in the speaker's knowledge of the world. Let we read a text in our native language on a topic that we know nothing about, but written for an audience that does know the topic. We may identify all the words and parse all the sentences, but have little idea as to what the author is saying. Without the particular knowledge that the author assumes of the reader, one cannot understand at more than a superficial level. Knowledge of the language is not enough; knowledge of the world is required.

Knowledge of the world is particularly important in the resolution of ambiguity and anaphors. Frequently, only one reading of an ambiguous sentence will make sense, or one will be more plausible or more likely than the others, given the appropriate knowledge. For example:

(7) George drank a glass of port.

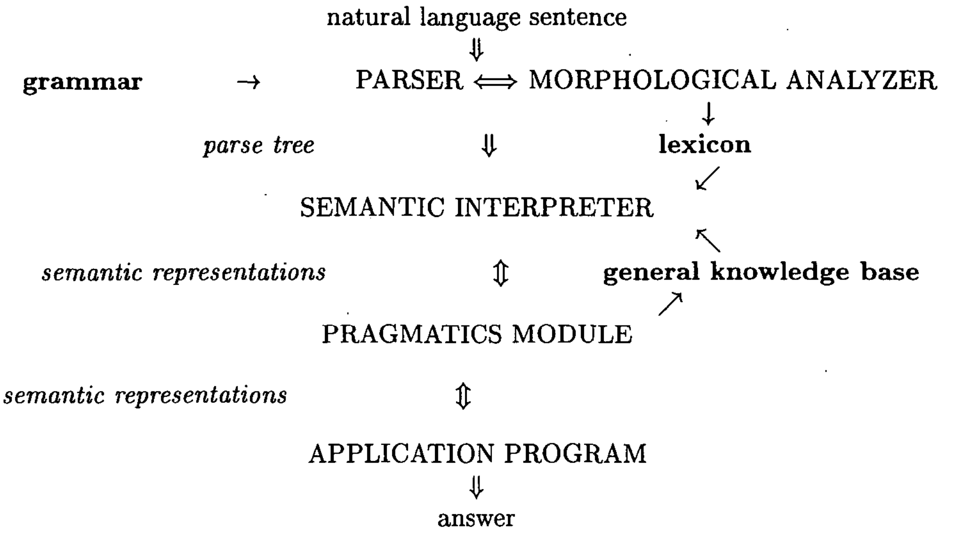
(8) George went to the library to get a book.

The word *port* can refer to a drink or to a certain place besides the sea. In order to interpret (7) correctly, we need to know from the world that only *port* as a drink can be put in a glass. Sentence (8) admits two plausible readings of *a book*: it could refer to a specific book that George is looking for or to any book. Both readings are plausible; thus, our knowledge of the world will help us to decide which to choose as probably intended by the speaker.

Although Bar-Hillel was the first, in 1960, to point out the need for knowledge of the world, its importance for natural language understanding has been only recently fully recognized. In the early days of computational linguistics, it was naively thought, for instance, that machine translation would require little more than a bilingual dictionary and a bit of morphological and structural analysis. Initial failures of such an approach were attributed to underestimating the complexities of syntax, and were unsuccessfully tried to solve by means of different kinds of syntax of increasing complexity.

4 The architecture of a natural language understanding system

The architecture of a standard natural language understanding system is as follows:



(Capital letters stand for processes and bold types represent knowledge sources. Arrows show the flow of information.)

The purpose of such a system depends on the application program, which could be, for example, a database system or a travel reservation system. It allows the user to easily ask in natural language, instead of having to learn some special formalism in order to interact with the computer.

A natural language understanding system shows a modular architecture, consisting of four major subsystems:

a) a morphological analyzer,

b) a parser,

c) a semantic interpreter, and

d) a module for discourse pragmatics.

Furthermore, the system has three main sources of knowledge:

a) a grammar,

b) a lexicon, and

c) a general knowledge base.

Finally, the application program uses to have its own specific knowledge base. The arrows are representative of the dynamic character of the system: the movement of information through it. The input is a natural language sentence, and the output is any form of answer from the application program. The answer could be in natural language, by means of a natural language generator. In between, the sentence is processed by each of the subsystems one after another, and then passed to the application program. As the arrows suggest, the subsystems do not act in an isolated way, but may interact to produce the final result.

Now, we are going to describe briefly each one of the subsystems and associated knowledge sources.

4.1 Morphological analyzer and lexicon

The lexicon is the list of words that the system has to recognize. The information listed for each word will typically include:

a) part of speech,

b) syntactic irregularities, and

c) representation of the meaning.

Irregular forms are usually listed as a cross-reference to the base form. If a word has more than one meaning or belongs to more than one part of speech, all are included. For instance:

(9) port = noun, regular; drink.
port = noun, regular; harbour.
port = verb, regular; to present (arms).

(10) men = plural, man.

The meanings shown are the names of knowledge representation structures where the detailed semantics of the words can be found.

The first thing that a natural language understanding system must do with each word it sees is to check that the word is in the lexicon. Normally, the lexicon will contain only the root forms of regular words, not plurals or inflected forms of verbs. If the word found is not in the lexicon, a morphological analyzer will try to determine the uninflected form. For many languages, this is a relatively simple matter of removing affixes and adjusting the spelling to see if the resulting word is included in the lexicon. In the case of agglutinative languages and others with complex morphology, however, the task may be complex and require a lot of interaction with the parser. The goal of this stage of the system is to discover all possible analyses of the input word. If it finds the word *drinks*, for example, it should report the possibility of a plural noun or a third-person singular verb.

If the system sees a word that it cannot find in the lexicon nor analyze morphologically, it must consider several possibilities:

a) that the unknown word is actually a known word mistyped: in this case, spelling correction techniques have to be attempted;

b) that it is a special word such as a bank account number, which could be the subject of queries to a business application: in principle, this kind of words can easily be recognized;

c) that the word is a name: the parser will have to determine whether a name could occur at this point of the sentence;

d) that it is a word which has been unconsciously omitted from the vocabulary of the system: then, if the system is interactive, the user will be asked to either rephrase the sentence without it or add it to the lexicon.

4.2 Parser and grammar

The parser is the component of the system that determines the syntactic structure of the sentence. The input to the parser is the sentence, and the output is a parse tree or phrase marker.

As it is building the tree, the parser draws upon information from the lexicon as well as from the morphological analyzer; if offered more than one possible morphological analysis of a word, it takes the one that best fits the context. And, of course, the parser needs to know the grammar of the language that it is parsing. Usually, the grammar is represented separately, in such a way that the parser can draw upon as it needs to. In theory, the parser can analyze any language whose grammar, morphology, and lexicon are given; it contains the universal and general principles of syntax, independent of any particular language. In practice, however, most real parsers that have been developed so far have been limited to at most a few typologically related languages (belonging to the same family).

There are many different kinds of parser, and many different ways of representing the grammar of a language. The two most common types in computational linguistics are:

a) chart parsers, and

b) augmented transition network parsers (ATN's).

A chart parser attempts to find a combination of words allowed by the grammar that matches the input sentence. This may involve much trial and error. A chart parser maintains a chart of the alternatives tried and the hypotheses tentatively accepted.

An ATN represents a grammar as a network; to parse a sentence is to traverse the network, respecting the constraints on each path (for instance, that the next word in the sentence must be a verb). If the parser finds itself unable to proceed, it must backtrack to some previous point and try another alternative.

Often, a sentence will be syntactically ambiguous; that is, the grammar will produce two or more different parse trees. For example, in:

(11) George is seeing John with the telescope,

the prepositional phrase *with the telescope* could describe the seeing, that is, complement the verb, or John, that is, complement the object noun. Deciding which one is intended by the speaker requires considering the meaning and relative plausibility of each. To find this out, the parser will have to ask the semantic interpreter (connected to the general knowledge module) about the meanings of the alternatives; thus, parsing generally alternates with semantic analysis.

4.3 Semantic interpreter and general knowledge base

The ultimate goal of the analysis is to determine the meaning of the sentence. The semantic interpreter needs not wait until the parser has completed its job; usually, it can begin to work on each constituent of the tree as soon as the syntactic analysis of that constituent is complete, regardless of the state of the rest of the analysis. Indeed, many systems rely on this possibility in order that semantics be able to assist syntax.

Meaning is represented in a computer system by means of knowledge representation formalisms or logics. The lexicon gives the meaning of each individual word in such a formalism, and the semantic interpreter must combine these in a manner appropriate to the structure of the sentence and the meanings themselves, either in strict accordance with the principle of compositionality or not. If a word conveys more than one possible meaning, as most words do, then the semantic interpreter must decide which one was intended by the speaker. Usually, this requires determining which makes more sense in the context. The result of this whole process is a logical form that represents the literal meaning of the sentence.

4.4 Pragmatics module

Computers tend to carry out literal interpretations. For example, a computer asked to:

(12) Give me the examination grades of all the mathematics students

might answer with just a list of anonymous marks. Humans often say things obliquely or incompletely, leaving to the intelligence of the listener to determine the intention and fill in the gaps. To be of practical use, a natural language understanding system must follow its literal semantic analysis with a pragmatic analysis, determining what the speaker really meant and how the sentence fits into the conversation.

Imagine that a user says to a travel reservation system:

(13) I've been thinking about going to Hungary.

At the literal level, the user is just only stating a fact that the system could merely take note of. But, at a deeper level, the user is asking the system to provide information about schedules of travels to Hungary. The system must recognize that the user is indirectly asserting that he/she has a goal of finding information about travelling to Hungary, and is asking for help in achieving that goal; that is, the computer has to recognize that the sentence is an indirect speech act. In order to determine the speaker's intention, the system must use not only knowledge of standard linguistic conventions, but also knowledge of how people plan and how their goals can be achieved. Thus, at present the problem of plan inference in natural language systems occupies an outstanding position in computational linguistics.

Just at the pragmatic level the system must also determine how the sentence relates to the preceding conversation or discourse. For example, it may exemplify or elaborate on the previous sentence, or describe the next in a sequence of events, or change the topic of conversation. Sometimes, speakers will make the relationship explicit: *for example* might be used to mark an exemplification, and *by the way* a change of topic. But often the relationship is left implicit and must be determined from the meaning itself of the sentences. This task can be quite complex for computers. Consider, for example, a sentence intended as a conclusion to be drawn from the preceding sentence, as in the following pair:

(14) Nobody likes the new tax system. The government is certain to be defeated.

The system must determine that the second sentence could plausibly be a consequence of the first one.

5 Parallelism in natural language processing

We have seen a decomposition of natural language automatic description into a series of different coordinated levels. Models of sentence processing may or may not refer to this decomposition. Natural language processing systems can be built for quite practical reasons, and therefore efficient performance properties can be much more important than attempting to reflect theoretical ideas coming from linguistics or psychology. Since practical systems do not always have to deal with the full range of natural language sentences –or with an unlimited domain of discourse– , the natural decomposition we have provided does not need to be explicitly present in language processing systems. From a psychological and linguistic point of view, however, computer models of human sentence processing should be consistent with theories developed in those fields. Having a model, it should be possible to simulate phenomena of human sentence processing.

Human sentence processing was initially explained by means of a serial model. This kind of models use a syntactic approach, where the syntactic processing task must be successful before semantic processing can begin, which in turn must precede pragmatic processing. If, in this model of linear interaction between levels of knowledge, higher-level information cannot be used to correct decisions at lower levels, this approach inexorably leads to a combinatorial explosion of all syntactic possibilities. For such reason especially, approaches combining different levels closely interacting at different moments are now preferred.

Models in which this latter type of sentence processing can be displayed are called interactive or parallel. During parsing, a system is capable of using any type of knowledge at any moment it needs. These models may exhibit different appearances. They can take, for instance, the form of a system in which natural language processing tasks are assigned to different processors and in which every knowledge source interacts with every other. In the known blackboard model of interaction, modules can process in parallel and cooperatively by means of a globally accessible blackboard on which they can write and read intermediate results: the modules communicate and interact solely through the blackboard. Some further possibilities exist.

6 Thesis

Assuming the natural decomposition of language we have shown and the parallel type of processing, we regard natural language as the final product of a parallel communicating grammar system (PC) architecture, each one of whose processors simulates one of the modules of natural language we have considered. In addition, each component of the parallel communicating grammar system consists of several subprocessors working as cooperatively distributed grammar systems (CD). We would have, then, a two-levels machinery: a macro-PC-system composed by micro-CD-systems. Its functioning would be as follows. Several processors cooperate distributively in the complex task of producing a syntactically well-formed (grammatical) sentence. Each one of such processors generates one of the levels we can distinguish in the syntactic structure of the sentence. On the other hand, it seems clear that human language is not produced/understood in a serial manner, but in a parallel one: syntax is not strictly generated before semantics can intervene, but in accordance with a complex synchronicity. Different levels and sublevels of each module of language are successively integrated in accordance with a certain protocol of integrative cooperation.

Computer scientists have now the task to formally define such two-levels machinery, and linguists the task to characterize the programme of synchronization of the modules. Both works are strong challenges for the future. If carried out jointly, the forecast is encouraging.

References

[1] Allen, J. (1987), *Natural language understanding.* Addison-Wesley, Reading, Mass.

[2] Csuhaj-Varjú, E. (1994), "Grammar systems: a multi-agent framework for natural language generation", in Gh. Păun, ed., *Mathematical aspects of natural and formal languages*: 63-78. World Scientific, Singapore.

[3] Csuhaj-Varjú, E. & R. Abo Alez (1995), "Multiagent systems in natural language processing", unpublished ms.

[4] Csuhaj-Varjú, E., J. Dassow, J. Kelemen & Gh. Păun (1994), *Grammar systems: a grammatical approach to distribution and cooperation.* Gordon and Breach, London.

[5] Dowty, D., L. Karttunen & A. Zwicky, eds. (1989), *Natural language parsing.* Cambridge University Press, New York.

[6] Grishman, R. (1986), *Computational linguistics.* Cambridge University Press, Cambridge.

[7] Grosz, B., K. Sparck Jones & B. Webber, eds. (1987), *Readings in natural language processing.* Morgan Kaufmann, Los Altos, Ca.

[8] Păun, Gh. (1995), "Generating languages in a distributed way: grammar systems", in C. Martín Vide, ed., *Lenguajes naturales y lenguajes formales, XI*: 45-71. PPU, Barcelona.

[9] Păun, Gh. (1995), "Grammar systems: a grammatical approach to distribution and cooperation", in Z. Fülöp & F. Gécseg, eds., *Proceedings of ICALP'95*: 429-443. Springer, Berlin.

[10] Păun, Gh. (1995), "Parallel communicating grammar systems. A survey", in C. Martín Vide, ed., *Lenguajes naturales y lenguajes formales, XI*: 257-283. PPU, Barcelona.

[11] Smith, G.W. (1991), *Computers and human language.* Oxford University Press, New York.