Heterogeneous Natural Language Understanding in Helios

Hiroshi TSUDA Akira AIBA
Institute for New Generation Computer Technology (ICOT)
{tsuda, aiba}@icot.or.jp

Abstract

Natural language understanding includes a variety of constraint domains, processing units, and processes such as constraint solving and relaxation. Moreover, now that data size to be processed is becoming larger, reuse of existing natural language resources is necessary. This paper applies Helios, a heterogeneous distributed cooperative problem-solving system developed in ICOT, to natural language understanding with communicating, especially negotiating, agents. At first, constraint-based grammars such as HPSG and JPSG are considered, where understanding corresponds to solving heterogeneous kinds of constraints such as unification, temporal logic and subsumption relation. Secondly, we take up a kind of constraint relaxation which occurs in garden-path sentence recognition, semantic/syntactic interaction, disambiguation, and ill-formed sentences in NLP. This paper gives a negotiation model for such phenomena.

1 Introduction

Seen from the view point of diversity, cooperation, and constraint processing, the following three aspects are important in natural language processing.

Firstly, natural language processing contains a variety of constraint domains and their processing units. In constraint-based grammar formalisms [11], most of the grammatical information is effectively and declaratively represented in the form of constraints. To explain various natural language phenomena, the constraint domain must be diversified, such as symbolic, temporal, and term unification. Linguistically, constraints are classified into syntax, semantics, pragmatics, and so on.

Secondly, seen from view of constraint processing, natural language processing involves various kinds of processes: not only constraint satisfaction but constraint relaxation. Usually, natural language constraints can not be processed in a fixed order. Constraints are referred in the opposite directions in parsing and generation. When there are interactions among different kinds of constraints, some constraints with lower priority are neglected or relaxed. For example, in recognizing ill-formed sentences, there is no solution that satisfies whole constraints but only preferred constraints are solved with less processing costs.

Thirdly, the data size utilized in natural language processing is becoming larger and larger as shown in recent electronic dictionaries and large corpora. There also already exist various natural language processing resources such as dictionaries, parsers, and constraint solvers. From an engineering point of view, it is preferable to combine those existing natural language resources.

As a candidate for the framework to tackle the first two aspects, a multi-agent system (MAS)[9] can be considered. A multi-agent system is composed of independent processing units called *agents* that solve partial goals. Agents negotiate with each other, that is, they communicate with each other in a certain protocol and finally come to an agreement. During negotiation, agents may have to relax their own goals.

Processing units such as parsers, generators, and constraint solvers can be seen as agents. Most natural language phenomena are explained by not only one kind of constraint but various heterogeneous domains of constraints. In addition, not all the constraints are satisfied but there may be a conflict between constraints, and less important constraints are relaxed or ignored due to their priorities and processing costs.

In ICOT, a heterogeneous distributed cooperative problem-solving system Helios [1, 2] is now under development. Helios can combine existing problem solvers such as DBs, constraint solvers, and application programs as agents in an MAS fashion. Helios can be seen as a framework that satisfies above three aspects required in natural language processing with heterogeneous constraints.

In this paper, we take up some phenomena in natural language understanding and explain them from an multi-agent view, illustrating how they are treated in Helios. The first example is processing of constraint-based grammar formalisms that consists of heterogeneous domains of constraints. In JPSG (Japanese Phrase Structure Grammar)[4], not only term unification but temporal constraints [5] are utilized to explain varieties of readings of Japanese verbs.

For the second example, we consider constraint interaction in some cognitive processes such as garden-path sentence recognition, semantic/syntactic interaction, and disambiguation. In Helios, those phenomena are realized as negotiation between heterogeneous constraint solvers.

2 Heterogeneity in Natural Language

2.1 Variety of the constraint domain

One kind of heterogeneity in NLP is the variety of constraint domains. In [12] Shieber pointed to several constraints in natural language in terms of various linguistic categories and case studies as follows.

Linguistic category	Case study	Constraint type
syntax	efficient encoding coordination	term equation, term inequations, subsumption
semantics	ellipsis	higher-order term equation
pragmatics	collaborative dialog	temporal constraint
	pronoun reference, translation	statistical constraint

Constraint-based grammar formalisms such as HPSG (Head-driven phrase structure grammar) [8] and JPSG (Japanese Phrase Structure Grammar)[4] describe natural language information in terms of phrase structure grammars whose nodes are feature structures. JPSG is a constraint-based grammar specially designed to treat Japanese. Until 1992, formalism and processing of JPSG had been discussed in PSG-working group at ICOT.

Figure 1 shows a simple treatment of subcategorization and agreement using a sentence "Ken walks." Each node is a feature structure that is a set of feature-value pairs.

Most of the grammatical information is stored in the form of local constraints between nodes in a phrase structure. The following is a PATR style representation[11] of a phrase structure and its grammatical constraints used in the example of Figure 1.

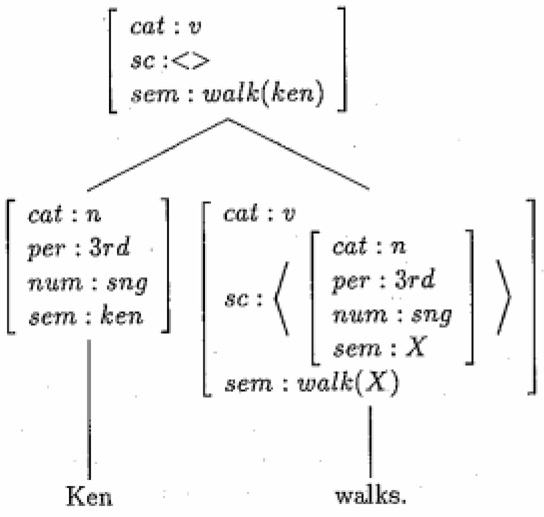

Figure 1: Simple treatment of "Ken walks."

$$
\begin{array}{lll}
M \rightarrow C\ H: & & \\
 & < M\ head >=< H\ head > & \text{(Head Feature Principle)} \\
 & < H\ subcat >= [C| < M\ sc >] & \text{(Subcat Feature Principle)}
\end{array}
$$

In Figure 1, *cat* and *sem* feature are called head features obeying the first constraint (Head Feature Principle): *The head feature of M unifies with H's.* The *sc* feature is called a subcat feature and obeys the second constraint (Subcat Feature Principle): *The first element of H's subcat feature unifies with C, and the rest are M's subcat feature.* Here, the constraint domain is a term unification. These constraints specify subcategorization and agreement information of English.

2.2 Variety of processing

Another kind of heterogeneity can be considered in the various processing units, such as lexical analysis, syntactic processing, and semantic processing. Many NLP system are constructed by connecting these modules linearly. Such a linear system, however, is too simple to mirror the diversified information flow in actual human language usage because the processing mechanism is fixed in advance. The constraint-based grammar noted earlier is also a framework that allows flexible information flow. As grammars are defined declaratively in terms of constraints, they are used in various directions such as parsing and generation.

There are several phenomena explained as a variety of processing in natural language understanding. In these cases, constraint relaxation as well as constraints solving must be considered. For example, sentences (1) are called *garden-path sentences*, because many readers backtrack to understand them[7]. In (1a), "cotton clothing" is read as a noun phrase at first, although the parse fails when one comes to "grows." In these cases, not all the constraints are solved; only partial constraints are solved depending on the processing cost.

(1) a. The cotton clothing is made of grows in Mississippi.
b. Have all the eggs broken {? or !}

Sentences (2) are examples of semantic/syntactic interaction[7]. They are all syntactically correct sentences. However, the acceptability differs in terms of syntactic and semantic preferences in constraints. (2a) is recognized as syntactically and semantically good, (2b) semantically bad, and (2c) mildly good.

(2) a. Which dragon did the knight give the boy?
b. * Which boy did the knight give the dragon?
c. (?) Which boy did the knight give the sword?

[7] explains those phenomena using preference in syntactic and semantic constraints.

The following sentences (3) are examples of disambiguation (parse preference). Although they are all syntactically ambiguous, each sentence has a preferred reading. The preference comes from syntactic processing principles, constraint priorities, constraint interaction, and so on. In (3a), the reading "John knows (that) the best man wins." is preferred to "John knows the best (thing that) man wins." The preference comes from a syntactic principle: disfavor of headless structure[6]. In the next two Japanese sentences, the first adjective phrase (ADJ) tends to modify the nearest following noun (left association principle) as shown in (3b). In (3c), however, "tsumeeri" (a stand-up collar uniform) and "joseito" (a girl student) are semantically inconsistent. Consequently the reading to modify the last noun ("gakusei") is preferred.

(3) a. John knows the best man wins. [6]

b. Se no taka-i joseito-wo suki-na gakusei.
tall-ADJ girl students-OBJ like boy student
"a boy student who likes tall girls"

c. Tsumeeri-no joseito-wo suki-na gakusei.
a stand-up collar uniform-ADJ girl students-OBJ like boy student
"a boy student in a stand-up collar uniform who likes girls."

Besides the above phenomena, ill-formed sentences, which are syntactically bad although semantically understandable, can be considered as inter-module constraints between semantic and syntactic processing.

3 Helios: Heterogeneous Distributed Cooperative Problem Solving System

3.1 MAS framework

What is a suitable framework to treat heterogeneity in natural language? The framework must have open constraint domains because we cannot predict which domains of constraint will be used in advance. Also the framework must be composed of various modules and flexible information exchange between them must be available.

As an candidate for such a framework, we consider the MAS (Multi-Agent System) [9] in DAI (Distributed AI). A multi-agent system consists of independent processing units called *agents*. Agents can negotiate with each other to solve an inter-agent problem.

Negotiation is realized with a negotiation protocol and a negotiation strategy. Negotiation protocol defines what message sequences are allowed. Every agent in the negotiation must follows the protocol. Each agent is characterized by its utility function and negotiation strategy. The negotiation strategy defines which message to reply to next and which information is contained in the message, so as to maximize the value of the agent's utility function.

3.2 Helios

In ICOT, a heterogeneous distributed cooperative problem-solving system called Helios is under development[1, 2]. Helios is used to form an MAS from existing problem solvers such as constraint solvers, DBs, and knowledge representation languages. In Helios, these solvers, called *substances*, are wrapped by *capsules* to become agents. Agents can communicate with each other by messages through the *environment* in which they exist. The environment defines a common message data (type) structure and several directories of inside agents to deliver messages to suitable agents. The capsule manages messages imported to and exported from the agent, and converts data between the environment and its substance.

In Helios, agents are assumed to have the following abilities:

- return the answer to a given problem and
- ask subproblems that it cannot solve to the environment and wait for the response.

For example, when an agent has subgoals that it cannot solve, it throws them out to its environment. Messages in Helios consist of a message type, a message identifier, destination specification, contents, a method name, and results.

In a message, a flexible destination specification is available, which uses an agent name and a function name as a label attached to a set of agents. From this destination information, the environment delivers the message to other suitable agents and the answers are returned to the sender agent. When there are more than one suitable destination agents, the sender agent can specify the way the results are handled as bag_of, taking the first answer, selecting only the results that meet certain constraints, and so on.

Helios supports negotiation by its transaction mechanism and negotiation protocol specification[2]. A sequence of messages is labeled by a transaction identifier. Negotiation protocol, which is defined in an environment, is a set of sequences of methods allowed in the negotiation.

The Helios system provides two languages, CAPL (CAPsule Language) and ENVL (ENVironment Language), to define a capsule and environment. CAPL defines method definitions, the self model of the agent, message translation between the common data structure and the local one, the negotiation strategy, and so on. On the other hand, ENVL defines global information such as the common data structure among inside agents, some directories used in delivering messages, global constraints, negotiation protocols, and so on.

4 Natural language understanding by cooperative agents

4.1 MAS View of Natural Language Understanding

In this section, we apply an MAS framework to explain several natural language phenomena. As introduced in Section 1, we focus on the following three aspects in natural language processing:

- variety of constraint domains and processing units,
- constraint satisfaction and relaxation among various kinds of processing, and

* reuse of existing natural language processing resources.

As an example of the importance of the first aspect, we take up an example in constraint-based grammar processing in 4.2. As examples of the second topic, several cognitive phenomena are explained in 4.3. The last aspect is already implemented in Helios, because Helios can make existing constraint solvers into communicating agents.

4.2 Processing constraint-based grammar

JPSG extends the framework of HPSG to treat Japanese particularly. JPSG postulates only a few phrase structure rules whose nodes are feature structures, and most of the grammatical information is stored in local constraints between the nodes in a phrase structure. Phrase structures are merely back-bones and most of the grammatical information is declaratively described as local constraints among nodes in phrase structures. As constraint-based grammar was called "unification-based grammar", JPSG was based on only the unification constraints [4] as shown in 2.1. To express various linguistic phenomena in Japanese, however, other domains of constraints are required. For example, temporal constraints are utilized to explain readings of tense and aspect in Japanese [5]. In (4), the verb "kite-iru" has three temporally different meanings.

(4) a. Ken-ha ima isshoukenmei sono kimono-wo kite-iru.
ken-SBJ now, intensively the clothes-OBJ wearing
"Ken is now putting on the clothes. (basic)"

b. Ken-ha kesa-kara zutto sono kimono-wo kite-iru.
ken-SBJ from this morning, continuously the clothes-OBJ wearing
"Ken has been wearing clothes from this morning. (resultative)"

c. Ken-ha sono kimono-wo sannen mae-ni kite-iru.
ken-SBJ the clothes-OBJ three years ago wearing
"Ken has the experience of wearing the clothes three years ago. (experiential)"

JPSG explains this variety by using temporal constraints in dictionaries and grammatical constraints. Each verb entry in the proto-lexicon is attached to the *tmp* feature that takes three parameters $< s, f, r >$, where s is the starting time of the event, f the finishing time, r the recovery time, and there is a temporal constraint $s \leq f \leq r$. In the working dictionary, the *tmp* feature is expanded to the *view* feature that takes two parameters $< b, e >$, where it is at most three-way ambiguous: $< b, e > = < s, f >$, $< sf, r > (sf = s = f)$, or $< sfr, \infty > (sfr = s = f = r)$.

Consider the implementation framework of such a grammatical formalism with a variety of constraints. To process JPSG, the author developed an CLP (extended Prolog) system to treat feature structure and unification constraints [13]. It may be possible to develop a big system to process various kinds of linguistic constraints. However, it is not a promising approach, because it is not flexible for the extension of grammar and does not reuse existing problem solvers.

In Helios, on the other hand, the JPSG parser is implemented with communicating problem-solver agents such as a CFG parser, a feature structure unifier, a temporal constraint solver, and so on (Figure 2). Various kinds of constraints are obtained from the following modules:

* dictionary: temporal constraints in the verb dictionary, disjunctive feature structure constraints from lexical entries of ambiguous words, and

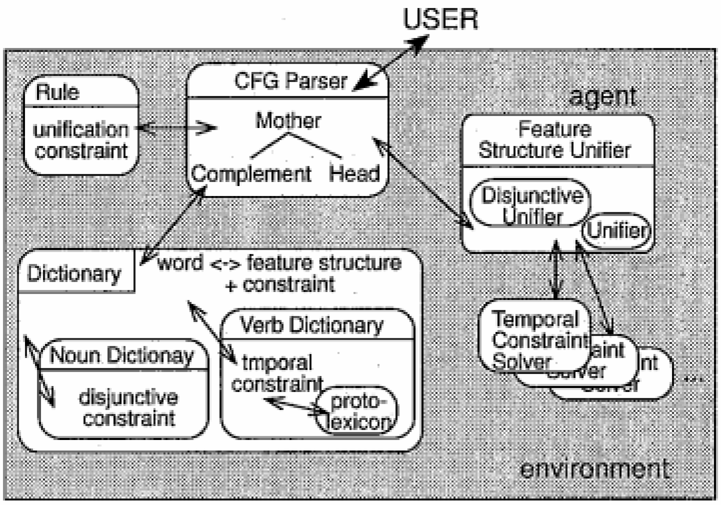

Figure 2: JPSG parser in Helios

- rule: grammatical principles in terms of unification constraints in phrase structures.

In the parsing process, the CFG parser agent acts as a coordinator. When the parser agent comes to the stage of consulting dictionaries, getting an appropriate CFG rule, and solving constraints, the agent throws the goal to the outside environment. The environment dispatches the message to appropriate constraint solvers and waits for the result, which will be sent back to the parser agent. In each constraint solver, constraints are transformed into a canonical form if they are satisfiable. The advantage of the constraint-based natural language processing approach is that ambiguity packing is automatically realized by the constraint solving[13].

4.3 Interaction between various constraints

As shown in Section 2, not all the constraints are completely solved but constraint relaxation is often necessary in natural language understanding. As the second example, we show several constraint interactions occurring in the cognitive process.

Consider semantic/syntactic interaction in (2) mentioned previously. In [7], the difference in their acceptance is explained as the following interaction between semantic and syntactic constraints.

- syntactic constraint:
  The noun phrase next to the verb tends to be IO (indirect object).
- semantic constraint: The order for an IO is higher_animate > animate > inanimate.

(2a) is good because the syntactically preferred reading is also semantically preferred, (2b) bad because the preference is opposite, and (2c) mildly good because only one reading is semantically acceptable.

How are those phenomena realized in a multi-agent framework in Helios? The following aspects must be considered.

- There is a priority to each constraint and the reading that meets the most constraints is preferred.
- There are processing costs and the reading with less cost is preferred.

Agents are equipped with local constraints with preferences and utility functions. Negotiation is done by exchanging messages in a certain protocol. Through negotiation, agents would like to come to an agreement that maximizes their utilities as much as possible.

In the case of semantic/syntactic interaction, there are two agents; a syntactic processing agent and a semantic processing agent. The utility function is the sum of preference values of satisfied constraints. Each agent also has a threshold of utility function; the goal whose utility is more than the threshold is acceptable. In the negotiation, the following messages are used:

Message	
plan(Plan_set)	Send a plan set to the other agent.
accept	Accept the other agent's plan.
eval(Plan,Value)	When the previous plan set is acceptable, return the most preferred plan and its utility value.
anti_plan(Plan_set)	Reject and send an anti plan.

The negotiation protocol is described as a state transition as follows. Here, s1 is a starting state and s4 an end state. The protocol is a modified version of [16].

State	plan	accept	eval	anti_plan
s1 : start	s2	-	-	-
s2	-	s4	s3	s2
s3	s2	s4	-	-
s4 : end	-	-	-	-

Example 1 *Semantic/syntactic interaction in negotiation: Sample negotiation to explain (2). SYN is a syntactic processing agent, SEM is a semantic processing agent.*

1. *Which dragon did the knight give the boy?*

   (SYN → SEM) plan([give(dragon,boy),give(boy,dragon)])
   (SEM → SYN) eval([give(boy,dragon)],1.0)
   (SYN → SEM) accept. (good sentence.)

2. **Which boy did the knight give the dragon?*

   (SYN → SEM) plan([give(dragon,boy),give(boy,dragon)])
   (SEM → SYN) eval([give(boy,dragon)],1.0)
   (SYN → SEM) plan([give(dragon,boy)])
   (SEM → SYN) eval([give(dragon,boy)],0.5)
   (SYN → SEM) accept. (semantically bad sentence.)

3. *(?) Which sword did the knight give the boy?*

   (SYN → SEM) plan([give(boy,sword),give(sword,boy)])
   (SEM → SYN) anti_plan([give(boy,sword)]) (SEM has only one acceptable plan.)
   (SYN → SEM) eval([give(boy,sword)],0.8)
   (SEM → SYN) accept.

In the classification by Rosenschein[10], WOD (Worth Oriented Domain) is appropriate for the above NL model, where each utility is defined as follows.

$$U_i = W_i(f) - c_i(f)$$

Here, U_i is a utility of agent i in the state f. $W_i(f)$ is the value of the the worth function of agent i in state f. $c_i(f)$ is the cost for agent i to come to state f.

In the example on semantic/syntactic interaction, W_i is the total of values of satisfied constraints. In the modeling of garden-path recognition (1) and disambiguation (3), c_i must be considered. The reading that satisfies as many constraints as possible does not always have high utility. As shown in the garden-path recognition, people first select the reading with less processing cost. The above form of utility function is appropriate to explain those phenomena.

5 Concluding Remarks

In this paper, we apply a multi-agent system framework Helios to natural language processing, showing treatment of constraint-based grammar, and semantic/syntactic interaction. This framework is flexible in the sense that any constraint solvers can be added to the system by extending the grammar formalisms. It also allows a variety of information flows using negotiation; not only constraint solving but constraint relaxation is implemented in the framework. Moreover, Helios has high reusability: many existent constraint solvers, dictionaries, and natural language processing units can be embedded in the system.

Hearsay (II)[3] is famous as a natural language processing system with distributed modules. Hearsay is a speech understanding system in which processing units such as speech input, a parser, and so on are connected via a blackboard. Ensemble[15] is a system to understand ill-formed sentences, in which a syntax-oriented parser and a semantic-oriented parser process an input sentence and the results are merged in a blackboard. Both of them are based on so-called blackboard architecture. Blackboard architecture in general, however, is not congenial to distributed processing. For example, in Hearsay[3] half of the processing time is overheads for the blackboard (read and write, synchronization, and so on).

In our approach, on the other hand, a global module such as a blackboard is not used, but flexible communication between modules (agents) is done using negotiation. Of course, communication overheads arise in our framework. The evaluation needs further study. If constraints do not interact so tightly, our approach can be effective compared to a blackboard architecture.

There are many natural language phenomena modeled by this framework. We are considering ill-formed sentences and treatment of situation dependency. The former is explained by negotiation between syntactic-processing and semantic-processing agents as shown in 4.3. In processing ill-formed sentences, there is no solution that is acceptable to both agents. Each agent have to revise the threshold to relax constraints during negotiation. This mechanism is similar to distributed constraint relaxation[14]. The latter occurs when the acceptance of a sentence differs according to the situation, and the preferred reading differs with the individual. Those phenomena can be modeled by situation dependency and individual difference of the worth function W_i in the utility of agents.

References

[1] A. Aiba, K. Yokota, and H. Tsuda. Heterogeneous Distributed Cooperative Problem Solving System HELIOS. In *FGCS94*, 1994.

[2] A. Aiba, K. Yokota, and H. Tsuda. Heterogeneous Distributed Cooperative Problem Solving System Helios and Its Cooperation Mechanisms. In *FGCS94 Workshop on Heterogeneous Cooperative Knowledge-Bases*, 1994.

[3] R. D. Fennell and V. R. Lesser. Parallelism in Artificial Intelligence Problem Solving: A Case Study of Hearsay II. *IEEE Transaction on Computers*, C-26(2):98–111, 1977. (also in Readings in DAI, Morgan Kaufmann, 1988).

[4] T. Gunji. *Japanese Phrase Structure Grammar*. Reidel, Dordrecht, 1986.

[5] T. Gunji. A Proto-Lexical Analysis of Temporal Properties of Japanese Verbs. In B.S.Park, editor, *Linguistics Studies on Natural Language*, pages 197–217. Hanshin Publishing, December 1992.

[6] J. R. Hobbs and J. Bear. Two Principles of Parse Preference. In *COLING Vol.3*, pages 162–167, August 1990.

[7] M. P. Marcus. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge:Mass, 1980.

[8] C. Pollard and I. A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications, 1994.

[9] J. S. Rosenschein. Consenting Agents: Negotiation mechanisms for Multi-Agent Systems. In *Prof. of IJCAI93*, pages 792–799, Chambery, 1993.

[10] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[11] S. M. Shieber. *An Introduction to Unification-Based Approach to Grammar*. CSLI Lecture Notes Series No.4. Stanford:CSLI, 1986.

[12] S. M. Shieber. Constraints and natural-language analysis. Tutorial in International Logic Programming Symposium, October 1991.

[13] H. Tsuda. cu-Prolog for Constraint-Based Natural Language Processing. *IEICE Transactions on Information and Systems*, E77-D(2):171–180, February 1994. (anonymous FTP from `ftp://ftp.icot.or.jp`).

[14] M. Yokoo. Constraint Relaxation in Distributed Constraint Satisfaction Problem. In *Proc. of ICTAI*, pages 56–67, 1993.

[15] A. Shimazu, K. Kogure, and M. Nakano. Bunsan kyouchou gata gengo rikai model no ganken na kaiseki heno ouyou (Application of distributed cooperative natural language understanding model to robust analysis). In *Workshop on Integration and Multi-modality*, JSSST, June 1994 (in Japanese).

[16] R. Kakei and M. Tokoro. A Negotiation Protocol for Conflict Resolution in Multi-agent Environments. In *MACC'92*, 1992 (in Japanese).