[From: *Current issues in computational linguistics: in honour of Don Walker*, ed. Antonio Zampolli, Nicoletta Calzolari, Martha Palmer (Linguistica Computazionale, vol. 9-10); Pisa, Dordrecht, [1994]

Varieties of Heuristics in Sentence Parsing*

Makoto Nagao
Dept. of Electrical Engineering, Kyoto University
Yoshida-honmachi, Sakyo, Kyoto, 606, Japan
e-mail: nagao@kuee.kyoto-u.ac.jp

1 Kinds of Grammars and Their Characteristics

There are many methods of sentence parsing, but parsing always presupposes a grammar, which is usually composed of a set of so-called grammatical rules or rewriting rules. There are many grammars proposed so far, and many parsing algorithms have been developed based on these grammars. Characteristics of these parsing algorithms are a direct reflection of the features of the grammar formalisms used by sentence parsing, so that we have to clarify the basic characteristics of these grammars.

We can classify grammars so far proposed into the following few classes:

(i) Phrase Structure Grammar (PSG)
Context-free PSG, Context-sensitive PSG, Augmented Transition Network Grammar, Definite Clause Grammar, Categorial Grammar, Lexical Functional Grammar, Generalized PSG, Head Driven PSG, Tree Adjoining Grammar,...

(ii) Dependency Grammar

(iii) Case Grammar

(iv) Systemic Grammar

(v) Montague Grammar

In the following we will discuss the basic ideas behind these grammars, and compare them contrastively from the standpoint of parsing.

1.1 Phrase Structure Grammar

Phrase structure grammar was proposed by N. Chomsky from the standpoint of sentence generation. This means that this grammar formalism is not necessarily fitted to the analysis of sentences. This will become clear when we consider the meaning of a

* This paper was presented at the 3rd International Workshop on Parsing Technologies, Tilburg/Durbuy, 1993, 8, 10-13, as an invited talk, but was not included in the proceedings.

grammar rule such as S → NP • VP. This rule declares that a sentence *should* be composed of NP followed by VP, or that a sentence *presupposes* the existence of NP followed by the existence of VP. This means that a sentence which is outside of this definition is excluded from the scope of the language. In this way this grammar formalism gives the definition of a language.

There is always a gap between an existing language and the set of sentences which a grammar can produce. The gap is not small, but actually very big. Sentences which we speak or write are essentially free, and cannot be grasped by such an artificial framework. We always encounter sentences or expressions which cannot be explained by a grammar, and we are forced to improve or add rewriting rules constantly. When a grammar is proposed as a tool to give a conceptual explanation of sentential structures of a language to a human being, a simple basic grammar will be sufficient. But when a grammar is to be used by a computer to parse existing sentences mechanically, there must be a very precise grammar, and its constant improvement will be required.

Japanese language is quite different from English and many other European languages. Japanese language is more free in word order change than these other languages. and there are varieties of omissions of essential components in a Japanese sentence. PSG has difficulty in handling these phenomena because the grammar formalism presupposes the word (phrase) order as is specified in grammar rules. Basically this grammar formalism does not fit languages which have free word order and where the notion of phrase structure does not hold.

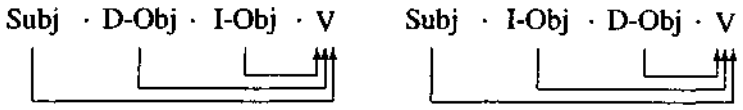
Similar discussion holds for ellipsis. When PSG is used for the analysis of a language which has varieties of ellipses, we have to try rule applications not only of the rules which have every component, but also of the rules which do not have (ignore) some of these components because it is quite difficult to specify under what condition a certain element can be omitted. This is almost impossible to execute. And the concept of grammatical restriction will not hold in such a case; that is, the phrase structure grammar will have no meaning any more.

PSG has such a serious problem in the analysis of sentences of at least a certain kind of languages such as Japanese. Nevertheless, many people use PSG for parsing without such considerations. We must see that there are other grammar formalisms which may be more suitable for sentence parsing.

1.2 Dependency Grammar

Dependency grammar (DG), which was treated in detail first by L. Tesniere is a grammar which is known as “Kakariuke” grammar in Japanese and which has been very popular for more than fifty years in Japan. This grammar is totally different from PSG in the sense that while PSG is a kind of language definition tool, DG is a kind of interpretation tool of a given sentence. DG clarifies which word modifies or depends on which other word. It constructs the modifier-modifiee relations between the words in a sentence, and this is always possible because a word in a sentence always has a relation to another word in the same sentence. It does not presuppose anything, but just clarifies the word relations in a sentence. In this sense DG can be regarded as a grammar for interpretation. The interpretation power of this grammar, however, is far weaker than that of PSG. DG does not say anything about subject, object, etc., but just say that this word modifies

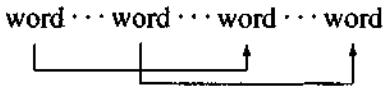
that, etc. However, this weak property becomes profitable, for example, for word-order change in Japanese because the modifier-modifiee relation is not influenced by the word-order change, as shown in the following:



DG is not concerned with the omitted words, and the analysis process is always the same.

So far so good. But from the standpoint of machine translation or some other natural language processing, the role of each word or phrase in a sentence must be clarified more accurately. In Japanese this is shown approximately by the information from verb conjugation, postpositions attached to nouns, and sometimes by the position of the word. So some additional analysis should follow DG analysis.

A serious problem in DG is an ambiguity problem—that a word can modify two or more words which follow the word in a sentence—and the decision is difficult. To solve this problem we have to prepare a good dictionary where the degree of affinity of two words is described. This is, however, not a particular problem for DG alone. It is a common problem for almost all grammars including PSG. This problem is relieved to a certain extent by the so-called non-crossing condition in the Japanese language, which can be illustrated by the following disallowed situation:



This condition is useful for the elimination of redundant checks of modifier-modifiee relation in Japanese.

1.3 Case Grammar

Case grammar (CG), which is proposed by C. Fillmore, is quite different from the above two grammar formalisms in the point that CG aims to clarify the roles of words in a sentence from the standpoint of meaning or of conceptual function of a word to a predicate. Therefore CG does not care about the word order in a sentence.

Case grammar interpretation of a sentence is represented by a case frame whose slots are filled in by words in a sentence. This case frame representation can be seen as a meaning representation of a sentence, so that it is comparatively neutral to a language. That is, sentences in different languages which express the same contents may have the same case frame. This is the main reason why many machine translation systems have adopted the case frame representation as the final goal of the sentence analysis and the starting point of sentence generation.

One of the difficulties of CG is that we have to tackle the difficult problem of formalizing the meaning representation because the grammar is based on meaning Fillmore did not explicitly discuss this problem because was not interested in machine processing but he relied on the human ability to interpret meaning. What he and his followers discussed seriously was what kinds of cases (slot functions in case frame) must be set up, such as agent, object, instrument, etc. What we, natural language processing researchers, had to do was to establish a semantic marker system which is powerful enough to distinguish different usages of verbs and nouns. Case frame must specify what kinds of nouns can be an agent, object, instrument, etc. of a particular verb in particular usage or meaning. The verb "kakeru" in Japanese, for example, has more than thirty different usage patterns, and the semantic markers should be detailed enough to be able to select the correct nouns for these usage patterns. It has been made clear by the effort of NTT researchers that 2-3 thousand semantic markers are necessary for the satisfactory description of every different usage of verb patterns (Ikehara 1991).

1.4 Other Grammar Formalisms

There are many grammar formalisms which can be classified into PSG, and these have the same problems which were discussed in Section 1.1. Montague grammar is based on formal logic, and the meaning of a sentence is represented by a logical formula. Present-day formal logic has a limited power of representation and it is impossible to express rich information of natural language in its limited formalism. There were several attempt at machine translation based on a Montague representation but these all failed because of the poor expressive power of logic and also of the difficulty of transforming a sentential expression to a proper logical form.

Systemic grammar, which M.A.K. Halliday proposed, is unique in the sense that it distinguishes three components in a sentence, namely ideational function, textual function and interpersonal function. Japanese language is rich in interpersonal functions. It has a sophisticated honorific expression system, for example, which uses specific words that affect sentential styles. Textual function corresponds roughly to discourse or contextual function. Systemic grammar is developed essentially for the generation of a sentence, and it is not clear whether it is useful to the analysis of a sentence.

2 Varieties of Heuristic Components in Parsing Sentences

2.1 Ambiguity Resolution in Sentence Parsing

In sentence parsing we are always confronted with the problem of ambiguity resolution of modifier-modifiee relation. To solve this problem we need such information as

(i) semantic consistency of modifier and modifiee.

(ii) word order and distance between modifier and modifiee. What kinds of words or symbols exist in between these two words are important for the rejection of the modifier-modifiee relation between the two words.

(iii) consistency with contextual/situational information.

(i) is checked by the consistency of semantic markers. To realize this all the words in a dictionary must be given proper semantic markers for their distinctive meanings. We may be able to utilize inclusion relations of semantic markers for the consistency checking.

(ii) is not easy to handle. Basically the modifier modifies the nearest possible modifiee, but it is not always true. We cannot specify a semantic relation so exactly as to accept just correct ones and reject the others. There always exist word pairs which cannot be accepted or rejected definitely. In these cases we have to look for other information to determine acceptance or rejection. It is often very useful to see words or symbols in between the two candidate words in a Japanese sentence. For example, the first word of a sentence which has -wa as a suffix usually modifies the last predicate of the sentence. But there are several cases where this condition does not hold. The following is an example:

Tokyo-wa	bukka-ga	takai-ga.	inaka-wa	yasui.
(Tokyo)	(price)	(expensive)	(countryside)	(cheap)

(Things are expensive in Tokyo, but cheap in the countryside.)

In this sentence there is another -wa and a predicate takai in between Tokyo-wa and the final predicate yasui, and these prevent the relation of these two words. In the following sentences,

Tokyo-wa	bukka-ga	takai-ga	hito-ga	atsumaru.
Tokyo-wa,	bukka-ga	takai-ga	hito-ga	atsumaru.
			(human being)	(gather)

the first sentence permits two interpretations, that is, Tokyo-wa modifies either takai or atsumaru (gather). But the second usually has one interpretation, that is, Tokyo-wa modifies atsumaru. This is caused by the comma after Tokyo-wa. We have developed a sophisticated algorithm to solve these modifier-modifiee problems (Kurohashi and Nagao 1992a).

As for case (iii) we have a famous example,

I saw a woman in the garden with a telescope.

Correct interpretation is possible only when we know the real situation of the utterance.

2.2 Anaphora and Ellipsis

When we proceed the analysis of a sentence from morphological analysis, syntactic analysis and semantic interpretation, that is, the transformation to case frame representation, anaphora and ellipsis are handled usually at the last stage of case frame representation. We can recognize that there is an ellipsis when we find out a vacant slot in a case frame representation of a sentence. We may be able to infer and recover this by utilizing contextual information so far obtained (sometimes we have to see some more words or sentences after this ellipsis position). Inference and recovery of a proper word is not easy, but we can write varieties of inference rules which check grammatical and semantic information requested from the case slot default information and which utilize the contextual information so far obtained.

As for the anaphora resolution, we can utilize grammatical information from pronominal words and write similar heuristic rules as above. We can of course think of another analysis process where anaphora determination is done just before the case frame transformation.

Looking for a proper word or concept for a pronominal reference or an ellipsis is - problem of language understanding. We have an example like the following.

Merikenko	to	Satou-o	yoku	maze,	atatameru.
(flour)	(and)	(sugar)	(well)	(mix)	(warm up)

where there is an omission of an object of the predicative verb, atatameru. When we trace back the sentence we encounter the nouns, Satou and Merikenko. These are the candidates for the omitted object. However, the actual thing to warm up is neither of the two. It is the mixture of these two materials, which does not appear explicitly in the sentence. We must introduce an inference mechanism that a mixture is created when two materials are "Mazeru-ed" (mixed). We have to have a mechanism to understand the meaning of a sentence and do the action which the sentence specifies, and get the result by applying an inference rule. In this way when there is a series of sentences which describe actions performed in time sequence, each action is supposed to be applied to the object which is produced as the result of the previous actions. We have to write many heuristic rules to produce these objects.

We may be forced to infer more than this. For the above example we cannot warm up the mixture of flour and sugar directly. The mixture must be in a certain thing such as a bowl. The determination of pronominal reference and the recovery of omission are quite difficult. We have to clarify what the language understanding is and what the common sense reasoning is.

2.3 Referential Property and Number

Besides the determination of pronominal reference and ellipsis we have to clarify the referential property (definite, indefinite, generic) and number (singular, plural, uncountable) of a noun in a sentence. In the Japanese language there are no indications such as articles and number suffixes in English to show these properties of a noun. Therefore we must estimate the referential property and number of a noun in a Japanese sentence. This requires language understanding by contextual information. It is difficult to achieve this at the present stage of research.

On the other hand we have to check how much information we can get to infer these properties of a noun from the sentence in which the noun appears. We are interested in this problem and tried to construct a kind of expert system to solve this problem (Murata and Nagao 1993). We wrote 84 heuristic rules to infer the referential property of a noun, and 48 heuristic rules to infer the number property.

Let us consider the following example.

Kinou	katta	piano-no	ichidai-wa	choritsu-ga	yokunai.
(yesterday)	(bought)	(piano)	(one unit)	(tuning)	(no good)

(One of the pianos which I bought yesterday is not tuned well.)

Piano in this sentence is modified by kinou katta (piano which I bought yesterday). So this piano is a concrete object which I have now, and the noun "piano" has the property definite. Piano is followed by ichidai, which means "one of • • •", so that we can infer that "piano" is not singular. Many such heuristic rules are written in the form of expert system rules.

The test of this system was done for two different sample sentence sets. The first test was done for the nouns in a set of sentences which were referenced in the process of writing heuristic rules. The success rate were 85.5% for the referential property and 89.0% for the number property. The second test was done for the nouns in a set of newly given sentences. The success rates were 68.9% and 85.6% for the referential property and the number property, respectively. This kind of analysis of nouns in a sentence is very important when we consider the construction of better machine translation systems.

2.4 Tense, Aspect, and Modality

We have to write many heuristic rules to interpret correctly the tense, aspect and modality of a predicate. These are particularly important in the interpretation of tense, aspect and modality of two or more predicates in a sentence when one of these is in an (i) embedded sentence, (ii) subordinate clause, or (iii) coordination. Many standard grammar books explain these relations in detail, but it is still very difficult to write expert system rules precisely for these relations.

Let us consider a pair of languages, for example English and Japanese. The categories and properties of tense, aspect and modality of Japanese are completely different from those of English. For example the Japanese language has no present perfect tense. This tense is often expressed by adverbs in Japanese such as follows.

Kare-wa ima tuita.
(he) (now) (arrived)
(He has just arrived.)

A more difficult situation exists where there are no such adverbs and we have to infer the tense from the situation.

Kare-wa Kyoto-e kita. Soshite ima kankou-o shiteiru.
(he) (Kyoto) (came) (and) (now) (sight seeing) (do)
(He has come to Kyoto. And he is now doing sight seeing.)

Kare-wa Kyoto-e kita. Soshite Kyoto daigaku-o sotsugyo shita.
(he) (Kyoto) (came) (and) (Kyoto University) (graduate) (did)
(He came to Kyoto. And he graduated Kyoto University.)

Sometimes different tense expressions in Japanese correspond to the same tense in English. The following is an example,

Sore-wa 10 nen mae-no koto-de aru.
(It) (10 years) (before) (matter) (is)
Sore-wa 10 nen mae-no koto-de atta.
(was)
(It was 10 years before.)

All of these problems are related to the problem of discourse. We don't know how many such problems there are and how many expert system rules we have to write The first problem we have to tackle will be to clarify how many different categories of discourse problems there are.

3 Stage Design for Parsing Sentences

3.1 Step-by-Step Analysis

As was mentioned in Section 1 each grammar formalism has its own characteristics, and so we have to choose carefully the best grammar formalism for a particular purpose. We believe that the following steps will be the best for the analysis of Japanese sentences for machine translation (Kurohashi and Nagao 1993).

(i) morphological analysis

(ii) detection of parallel structures

(iii) dependency analysis

(iv) case frame analysis

(v) textual function analysis

(vi) interpersonal function analysis

We will be able to design the analysis stage in different ways, such that the dependency analysis is skipped and the case frame is obtained from the result of morphological analysis, or that the total process may be merged into one by the constraint programming methodology and the final result is to be produced from the input sentence. However we don't recommend such processes. We believe that the best way is to divide the whole process into many subprocesses, and to perform small transformations at each subprocesses. The reason is that the analysis is an information losing process and it must be done very carefully in small steps so that important information will not be lost by a drastic change. There is an additional advantage that when the process is divided into many stages we can understand the details of each stage more easily and we can do the improvement of each stage independently.

3.2 Detection of Parallel Structures

Almost all the current parsing systems fail in the analysis of a long sentence, for example, a sentence composed of more than thirty English words or more than seventy Japanese characters. Nobody has analyzed the reasons for this difficulty. Probably very many factors are mixed up and the combinations are enormous, all of which are equally possible in causing errors in the analysis. There is no one major reason for the failure. We have to improve almost all the parts of the parsing process, particularly grammar rules.

On the other hand, we have to pose a question: Why is a sentence so long? The main reason is that people write a long sentence by connecting phrases, clauses and

sentences into one. Therefore an important point in the analysis of a long sentence is to find out such conjunctions. Serious efforts have been done to this problem so far by writing many grammar rules which check something like semantic similarities which may exist in the head nouns or main verbs of conjunctive structures. But there has been no significant improvement so far. This indicates that we have to think about a completely new approach to finding parallel structures in a long sentence.

What we have recently developed is based on the assumption that parallel phrases/ clauses /sentences will have certain similarities in the sequence of words and their grammatical structures as a whole. We had to compare two word-strings of arbitrary lengths from these aspects and to get an overall similarity value. Because we had to compare word-string pairs of arbitrary lengths in a sentence we adopted the dynamic programming method to calculate overall similarity values for all the possible word-string pairs of arbitrary lengths and to get the best one (the details are given in the paper Kurohashi and Nagao 1992b). The result was unexpectedly good. This algorithm has achieved more than 90% success rate in finding parallel structures of different kinds, and many long Japanese sentences which were composed of more than 100 characters were successfully analyzed.

The idea behind this algorithm is just to find out similar word strings in a sentence, which has nothing to do with the existing notion of grammatical rules. It is closer to human cognitive action which is vague, but which is very reliable in the global recognition process. The human brain may not work like the application of grammatical rules, but may work like the similarity detection mentioned here.

This algorithm was inserted in between the morphological analysis and the dependency analysis in our parsing system, and achieved a very good performance in sentence parsing.

3.3 Dependency Analysis and Conversion to Case Frame Representation

After the detection of conjunctive structures, the dependency analysis is first done to these parts, and then to the whole sentential structure. By adding this parallel structure detection algorithm the accuracy of the dependency analysis has become very high.

The transformation of the dependency tree of a sentence to a case frame representation is not difficult because the noun-verb modifying relations have been obtained at the stage of the dependency analysis. The work to do at this stage is to check in which case slots those nouns will come (Kurohashi and Nagao 1993). If there remains a vacant case slot after the assignment of words to suitable case slots, it is judged as an omission.

3.4 Recognition of a Phrase Unit

People utter a sentence not word by word, but phrase by phrase. This phrase unit is what Margaret Masterman called a breathgroup. This phrase unit has a unique meaning although each word which is a component of a phrase may have several meanings. This phrase unit is quite different from that of PSG in the sense that a phrase in PSG is hierarchically recursive, but a phrase here is not.

Example-based machine translation, which has been recognized as giving better quality translation than the other methods, memorizes lots of example phrases and

their translations as pairs, and a target language sentence is composed of the phrasal translations. Example phrases in this case are just this phrase unit which corresponds to the breathgroup. There is a discussion in machine translation study about the translation equivalence unit. Words have many meanings and are very ambiguous, so that they are not good for the translation equivalence unit. Sentences have structural ambiguity and are also too big to be an equivalence unit. Therefore phrases are good candidates as translation equivalence unit.

Nowadays there are lots of efforts to collect typical example phrases, and to construct a phrasal dictionary, because this dictionary contributes very much for the improvement of translation quality in machine translation. So recognizing phrases properly in a sentence has become an important task, particularly in example-based translation. When the phrases in a sentence are correctly recognized the analysis of a whole sentence becomes easy because the relations among these phrases are not so difficult to determine

4 Conclusion

We have discussed the essential properties of different grammar formalisms, and suggestioned that a proper grammar must be chosen for a particular purpose. For example generative grammar is not suitable for sentence analysis, so that the idea of bi-directional grammar, which aims at using the same grammar for analysis and synthesis of a sentence in machine translation, is to be reconsidered.

A sentence includes lots of information, such as syntactic information, semantic information, textual or rhetorical information, interpersonal information, and so on. We do not have any formal methods to detect this information. Even at the level of syntactic information PSG and other grammar formalisms are just a framework to explain basic structures of a language. No one knows how the human brain works to speak and recognize a language. We can just approximate the human language mechanism to a certain extent from various aspects. All are heuristic. Grammar formalisms, even PSG. are a kind of expert system. We have to write many expert systems to approximate a language at the levels of morphology, syntax, semantics, discourse, etc. Therefore we have to consider a parallel or pipeline execution structure of these expert systems because the total system is too big and complex to be executed sequentially. This will be a very interesting software problem in the near future.

References

[1] Ikehara, S., et al., "Semantic Analysis Dictionaries for Machine Translation", *IPSJ-NLP 84-13,* July 19, 1991 (in Japanese).

[2] Kurohashi, S. and M. Nagao, "A Syntactic Analysis Method of Long Japanese Sentences based on Conjunctive Structures' Detection", *IPSJ-NLP 88-1,* March 12,1992 (in Japanese).

[3] Kurohashi S. and M. Nagao, "Dynamic Programming Method for Analyzing Conjunctive Structures in Japanese", In *Proc. of COLING '92,* July 23-28,1992.

[4] Kurohashi S. and M. Nagao, “Structural Disambiguation in Japanese by Evaluating Case Structures based on Examples in Case Frame Dictionary”, In *Proc. of IWPT* '93, August 10-13, 1993.

[5] Murata, M. and M. Nagao, “Determination of referential property and number of nouns in Japanese sentences for machine translation into English”, In *Proc. of TMI '93,* July 14-16, 1993.