Are the grammars so far developed appropriate to recognize the real structure of a sentence?

Makoto Nagao
Department of Electrical Engineering
Kyoto University, Kyoto, Japan

Abstract

Some shortcomings of the present grammar formalisms are discussed, and the importance of checking wider scope of word arrangement in a sentence is stressed. The paper explains three research results of the author from this standpoint. One is the tree-to-tree transformation mechanism in machine translation process. Another is the detection of two similar word sequences in a sentence which can be recognized as a parallel structure. The third is the mechanism of a proper translation word selection. These are all supported by the principle of seeing wider scope of word arrangement and give very successful results.

1 Shortcomings of grammars so far developed

1.1 There exist many grammars which have been developed for the analysis of natural language sentences by computer. However none of them has been satisfactory for the analysis of real existing sentences, not even considering conversational sentences which are more irregular and sometimes ungrammatical. In the computational linguistics a major trend of grammar formalism was to merge two elements into one, typical one of which is the categorial grammar. Many grammars utilizing the idea of unification adopted this grammar formalism. However, this grammar formalism produces too many alternative analysis results, almost all of which are grammatically inappropriate in the sense of ordinary human linguistic intuition, and of course meaningless from the standpoint of semantics. Such a difficulty arises from the reason that this grammar formalism sees only two element relation at a time by ignoring that there exist many sophisticated relations among many elements in a sentence. To escape the shortcomings of this grammar formalism, unification grammar has attribute-value pairs for each grammatical component, and the unification operation rejects unmatched combinations of two grammatical components, thus avoiding redundant/meaningless analyses of a sentence. To realize this unification operation truly meaningful very many precise semantic markers are to be introduced which perform exact selectional restrictions for every sentential component. However, people who are interested in the unification grammar formalism are not making efforts in this direction. They are more interested in efficient algorithms of unification parsing, in parallel parsing and so on. They just handle the attributes such as singular/plural, male/female, human/non-human, etc., and nothing more sophisticated semantic markers. They are essentially people on software side, not on linguistics side.

1.2 There is another group of people who are interested in sentence parsing by case grammar formalism. They put on weight more on semantics than syntax in the analysis of a sentence. They claim that people understand ungrammatical sentences very often by the help of semantic coherency. Case frame information in the dictionary is the unique, most valuable information for them to analyze a sentence. Therefore when the case frame information is lacking for a verb in a sentence, the sentence cannot be analyzed properly.

Language has the ability to describe new contents and new concepts. They are explained by new combinations of known concepts (words). Because they are new, these combinations have not existed in a dictionary before. However people read and understand those sentences which convey new information. The reason why people understand new concepts is primarily by new combination of words, that is, by syntax. Even if an expression is semantically anomalous, if it is syntactically interpretable, then people are forced to accept the expression and interpret it semantically as it is. For example, "a stone flies in the air" is semantically strange. Nevertheless people accept it as it is. Metaphor utilizes this function and forces people to find out a deep reasonable interpretation.

Syntax is thus the basis in the interpretation of a sentence, and semantics plays the secondary role. It functions by the semantic preference principle when there exist multiple syntactic interpretations for a sentence. Moreover, we do not have any reliable and precise semantic theories at all which can be used in wide varieties of expressions. Even the largest dictionaries such as OED cannot be sufficient for the interpretation of delicate expressions. A language is always changing, and introducing new semantic relations among words.

1.3 Now, we have realized the primal importance of syntax rather than semantics in the interpretation of a sentence. But how can we avoid the mistakes the unification grammarians have made? Their mistake was that they saw only a very local part of a sentence at a time. I believe that we have to see many sentential components at a time. That is, we have to see more sophisticated grammatical phenomena in a wider range of a sentence than only two components at once. Unification grammarians may insist that the repetitive applications of two elements merge is equivalent to many elements merge at once. This is not true because the selectional restriction can be expressed more accurately by many-to-one merging rules than two-to-one merging rules, although the former needs very many rules than the latter. Linguists also prefer to represent collocational phenomena of several far apart elements easier in many-to-one merging rules than two-to-one rules.

2 To represent complex linguistic phenomena

2.1 We introduced the grammar formalism of tree transformation into our machine translation system which was built by the Japanese Governmental Project, called Mu Project, which was conducted during 1982-1986[l]. We developed a software system called GRADE which accepted a tree composed of any number of grammatical terms and converted it into another tree[2]. Each term could have attribute-value pairs, and semantic restriction could be expressed by using this information. Not only the attribute-value pair information could be inherited, but also some synthesized information from the attribute-value pairs could be given to a term in the transformed tree.

This tree transformation grammar formalism was used not only for the analysis but also for the transfer and generation of sentences in our machine translation system. Case frame information was given by this formalism, which generally included three to five grammatical components. In the transfer phase from Japanese into English we wrote the conditions for lexical selection by using this formalism. Sophisticated programs for the checking of varieties of conditions explicitly and implicitly existing in a tree structure could be added to each tree-to-tree transformation rules. For example we could perform the following transformation in the generation phase. A typical sentential pattern, "S V O_1 O_2" is sometimes forced to take another pattern "S V O_2 to O_1", particularly when the number of words which compose O_1 is big and that for O_2 is relatively small. Grammar writers in GRADE can use the checking function which counts the number of components which compose the phrase O_1 in a "S V O_1 O_2" pattern and can select either of the two expressions of "S V O_1 O_2" or "S V O_2 to O_1" according to the number of words in O_1.

2.2 By the existing grammar systems for machine translation it is quite difficult to get correct analysis for Japanese sentences of more than 70 characters and for English sentences of more than 30 words. The reasons are manifold. But one of the reasons is that too many meaningless analysis results come out by two-to-one merging rules, and that the analysis process becomes too complex to manage, and finally the process goes to failure, or strange results come out.

When we examine long sentences we find out many parallel structures such as conjunctive noun phrases, conjunctive sentential phrases and so on, which cause the analysis failure very often. Therefore the detection of parallel structures in a sentence *is* quite important. If we can succeed in it, the remaining structure of a sentence becomes very simple to parse, and very few multiple analysis results come out. In Japanese "Renyou Chushi-ho" is very often used to combine two or more sentential components into one sentence. A typical such expression is

NN_1 V_1-shite, NN_2 V_2-suru.
(do V_1, and) (do)

In this case V_1-shite can be paralleled with V_2-suru, and also can be paralleled with verbs which appear inside NN_2 portion as embedded sentential predicates. This causes the analysis multiple and complex. It is quite difficult to decrease the number of multiple

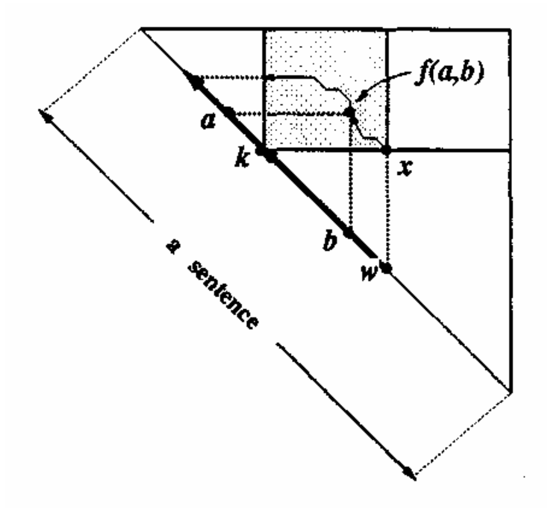

Figure 1: Dynamic programming method to recognize a parallel structure in a sentence

analyses because almost all arbitrary sentences can be connected by Renyou Chushi-ho, and no effective semantic restrictions can be imposed on V_1 V_2 pair.

We studied parallel structures of Japanese sentences in detail for scientific and technical texts, and found out a very interesting phenomena. That is, two componential word strings have similar syntactic structures when they form a parallel structure. It is not the simple similarity of head words in two componential word strings, the similarity of total phrasal structures of both componential word strings. If we apply the ordinary parsing strategy by using ordinary grammar rules to find out this kind of parallel structures we will fall into the same mistakes of wrong parsing or parsing failure, into which the parsing systems of current machine translation systems fall.

We introduced here a very new idea to find out similarity between two arbitrary word strings by using the dynamic programming method[3]. The method is shown conceptually in Fig. 1. An example is given in Fig. 2. We place a sentence on a diagonal edge of a triangle by word unit (in case of Japanese, by bunsetsu unit which is composed of a content word and suffix words). Then we find out in these units suffix words or special symbols which indicate possible existence of parallel structures on both sides of them. "to(and)", "matawa(or)", "," etc. are such typical words and symbols, and hereafter we call them keys for parallel structures. In Fig. 2 keys are indicated by > such as a>, b> and c>. In each crossing position of two words in the triangle of Fig. 1, the similarity value of the two words are given (in Fig. 1, for two words *a* and *b*, their similarity value *f(a, b)* is given to the crossing point). This is calculated roughly as follows. When the two words have the same part of speech, give score 2 (*f(a, b)* = 2), otherwise give score 0. Then if the two words are exactly the same, add score 10 (*f(a, b)* = *f*(a, *b)* + 10). If the two words are close in the thesaurus tree then add the score 2 ~ 10 according to the closeness, and so on. Then we start the similarity calculation of two word strings of arbitrary length, one starting from a key *(k* in Fig. 1) and going to the left, another starting from an arbitrary word *(w* in Fig. 1) which is in the right of the key and going to the left (see Fig. 1). Dynamic programming method is used to obtain the best pass inside the hatched square in Fig. 1 starting from the point *x* and going to the upper left. The best pass is obtained in the following way. Starting from point the *x* the addition of similarity values is performed when a step path is diagonally to the upper left, and no addition is done when a step path is horizontal or vertical. This process is done for all the possible paths from *x* until the pass reaches a grid point on the vertical line through *k*, and then the maximum addition value and its corresponding path are recorded at the position *x* (This is the dynamic programming method). This process is performed for all the words *(w)* to the right of *k.* Then the word *w* whose *x* has the largest value among all the possible words *w is* regarded as the headword of the second word string of the parallel structure. The leftmost word of the first word string of the parallel structure is the one which corresponds to the upper-left end of the best path from *x.* In Fig. 2 the best chosen paths for keys, a, b and c, are shown in the triangle by attaching the symbols a, b and c to the similarity values.

We achieved about 80% success rate in finding varieties of parallel structures which exist in long Japanese sentences. This score has never been achieved by ordinary syntactic analysis methods which parse only single word pairs. This experiment shows how important it is to see global structures of a sentence at once. Once these parallel structures in a sentence are recognized, the dependency analysis of a Japanese sentence becomes very easy. We achieved 96% correct recognition rate for the modifier- modifiee relation in complex long Japanese sentences[4]. An example is shown in Fig. 3, which is derived from the analysis result in Fig. 2.

2.3 In machine translation the stage of the selection of proper translation word

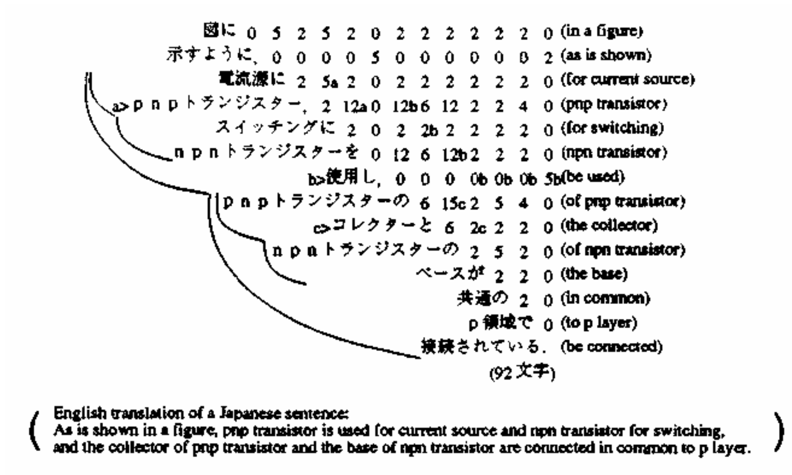

Figure 2: An example of analyzing parallel structures in a long sentence.

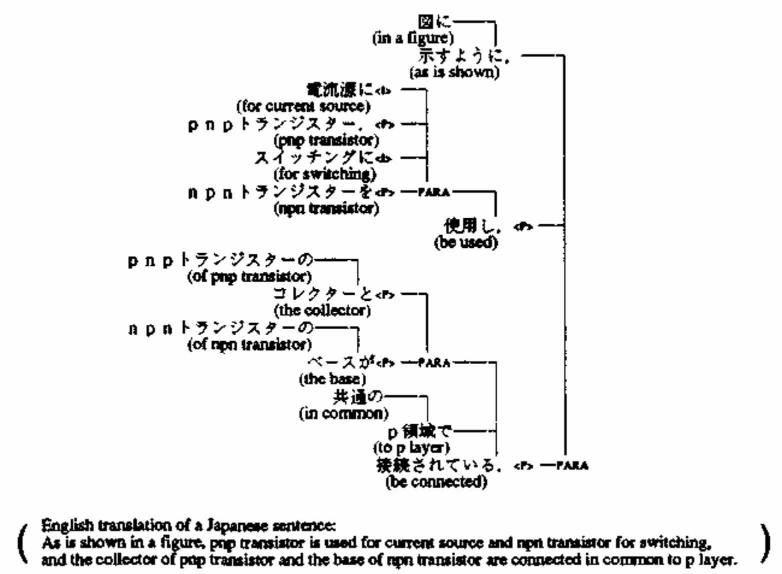

Figure 3: An example of analyzing a long sentence into a dependency structure.

comes next to the analysis stage. It is widely recognized that a contextual information is necessary for the selection of proper translation word, but nothing more details have been clarified yet for what kind of information is really required. A typical method at present is the utilization of case frame information in such a way as

$$O_J(\text{att. val.}) + V_J(\text{att. val.}) \rightarrow V_E(\text{att. val.}) + O_E(\text{att. val.}).$$

That is, when a Japanese objective noun O_J and a Japanese predicative verb V_J have such and such attribute value pairs, then an English noun and an English verb of such and such attribute value pairs are chosen correspondingly. This indicates that the contextual information is the semantic collocation of two words.

However, it is easy to show that this contextual information is not adequate. For example,

(Japanese)	(English)
coat-o ki -ni kakeru.	hang a coat a tree.
(tree)	
coat-o kata -ni kakeru.	put a coat on the shoulder.
(shoulder)	
coat-o shinda hito -ni kakeru.	cover a dead man with a coat.
(dead) (man)	

These examples show that at least three or more elements are closely related each other for the proper selection of an English word to the predicate "kakeru". "kakeru"s in the above three examples are regarded as the same word in the Japanese language. We can say in general that the collocational combination of several words in a source language determines one precise concept which can only be expressed by the combination of several words in a target language [5]. That is, the correspondence is a phrase to phrase from one language to another. This correspondence process is again expressed by tree to tree transformation in our machine translation system.

2.4 When we extend this idea of seeing wider range of word sequence in a sentence for translation, we come to the idea of example-based translation. Example-based machine translation was first proposed by the author in 1984 by the name of analogy-based translation [6]. The idea is the following. We accumulate lots of translation pairs of sentences and phrases of English and Japanese, and compare these with the given input sentence. There will be matches at the level of words, phrases or even at sentence level between the input sentence and accumulated example pairs. Then we obtain target language words, phrases or a sentence from the matched example pairs, and then we can synthesize these into a proper sentential structure of the target language. The quality of translation will be improved according to the richer accumulation of example translation pairs and the best match algorithm between two word strings. The method is not based on syntactic and/or semantic information directly, but is based on translation pairs which are always reliable. Therefore the method gives reliable results and also the translation quality can be improved very easily by increasing translation pairs. Therefore the method is adapted to machine learning.

3 Conclusion

As are shown in the previous examples it is absolutely necessary to see varieties of sentential components in a more and more wider range of a sentence for the proper analysis, transfer and generation in machine translation and other natural language processing systems. Grammatical rules so far developed in computational linguistics are only those which are definite, such as "subject + verb", etc., and did not give any considerations to such delicate phenomena as are explained in sections 2.2 and 2.3. A grammar is not restricted to the phrase structure or case grammar formalisms, but can be anything including such sophisticated mechanisms as the finding of parallel structure relations which is explained in section 2.2.

Modern western scientific methodology was based on the principle of divide and conquer. That is, to understand one whole thing, first divide it into more basic components and understand these. The repetition of this process goes until there is no more detailed componential elements at all, and there is no unknown element in the decomposition. The method is very powerful and has achieved the present-day scientific world. However we are gradually realizing that this principle is not appropriate or not applicable to social phenomena and also to many language problems. In such fields we cannot reconstruct the original whole from the divided elementary components by recognizing and utilizing the relations among these components. We have to have a new mechanism which can grasp the total structure as a whole. Nobody knows what it is. But many people feel that such a new mechanism is absolutely necessary. I don't know what it is, but I think that seeing more and more elements and factors in a sentence at the same time leads to the mechanism which grasp the whole thing at once. There has been a great progress in discourse theory which also handles very sophisticated implicit elements and mechanisms latent in wider range of a text and dialogue. This is another way to go, and the computational linguistic realization of the theory is to be investigated.

People may say that a theory must be simple and beautiful, and that it must be applicable to all the situations uniformly. In natural language area a basic theory may explain varieties of linguistic phenomena, but unfortunately in the first order approximation. The theory, however, cannot explain all the linguistic phenomena in the second order or the third order approximation, so to speak, and also there are many and many exceptions to the theory. What is called a theory in linguistics and computational linguistics is actually not a theory such as of mathematics and physics, but is merely a framework of hypothesis. We have to see more in detail the real language facts and to construct a better framework which covers more and more language phenomena. We have not to stay in the framework of a simple theory and enjoy discussions in the theory without considering language phenomena which cannot be handle by it.

References

[1] M. Nagao, J. Tsujii, J. Nakamura: The Japanese Governmental Project for Machine Translation, Computational Linguistics, Vol. 11, No. 2-3, April-Sept. 1985, pp.99-110.

[2] J. Nakamura, J. Tsujii, M. Nagao: Grammar Writing System(GRADE) of Mu-Machine Translation Project and its Characteristics, Proc. Coling 84, Stanford, 1984, pp.338-343.

[3] S. Kurohashi, M. Nagao: Dynamic Programming Methods for Analyzing Conjunctive Structures in Japanese, Proc. Coling 92, Nantes, 1992 (forthcoming).

[4] S. Kurohashi, M. Nagao: A Syntactic Analysis Method of Long Japanese Sentences based on the Detection of Conjunctive Structures, Submitted to Journal of ACL.

[5] M. Nagao: Some Dictionary Information for Machine Translation, Cheng-ming Guo, ed. Ablex Publisher(forthcoming).

[6] M. Nagao: A Framework of a Mechanical Translation between Japanese and English by Analogy Principle, A. Elithorn, R. Barnerji, eds. Artificial and Human Intelligence, North-Holland, 1984, pp.173-180.