Towards a Proper Linguistic and Computational Treatment of Scrambling: An Analysis of Japanese

Sandiway Fong*
NEC Research Institute, Inc.
4 Independence Way, Princeton NJ 08540. USA
sandiway@research.nj.nec.com

Abstract

This paper describes how recent linguistic results in explaining Japanese short and long distance scrambling can be directly incorporated into an existing principles-and-parameters-based parser with only trivial modifications. The fact that this is realizable on a parser originally designed for a fixed-word-order language, together with the fact that Japanese scrambling is complex, attests to the high degree of cross-linguistic generalization present in the theory.

1 Introduction

During the past several years, the phenomenon known as "scrambling" has become a topic of some interest; it is of particular importance in languages like German, Japanese, Korean and Hindi among others, as opposed to fixed-word-order languages like English. Scrambling can pose both severe linguistic and computational difficulties for natural language parsers. This paper describes how these problems are dealt with in a Principles-and-Parameters-based parser. Japanese, at first glance, seems to permit fairly-free permutation of objects:

(1) Short distance (VP-internal) scrambling

(a) John-ga Mary-ni kono hon-o ageta (koto)[1]
John gave this book to Mary

(b) John-ga kono hon-o Mary-ni ageta (koto)

(2) Short (or medium) distance scrambling to IP

(a) Mary-ga John-ni sono hon-o watasita (koto)
Mary handed that book to John

(b) Sono hon-o John-ni Mary-ga watasita (koto)

(c) John-ni sono hon-o Mary-ga watasita (koto)

(3) Long distance scrambling

(a) John-ga Mary-ga sono hon-o katta to omotte iru (koto)
John thinks that Mary bought that book

(b) sono hon-o John-ga Mary-ga katta to omotte iru (koto)

(c) Mary-ga John-ga Bill-ni sono hon-o watasita to omotte iru (koto)
Mary thinks John handed that book to Bill

(d) Bill-ni sono hon-o Mary-ga John-ga watasita to omotte iru (koto)

(Example (1) is taken from (Tada, 1993), and (2) and (3) from (Saito, 1985).)

To handle examples like these, computational linguists have sometimes adopted the straightforward strategy of adding permutation machinery on top of an existing formalism: for example, Becker et al.(1990) augment the Tree Adjoining Grammar (TAG) system using either: (1) multi-component (set-based) adjunction (MC-TAG), or (2) relaxed linear precedence (FO-TAG), to handle so-called "long distance" scrambling in German (that is, scrambling over clausal boundaries).[2] This augmentation aims to directly repair the lack of permutation in ordinary TAGs by *adding* a mechanism that can (over)generate many different scramblings. However, as so often happens, when one turns to a richer set of examples in other languages, or the interaction of scrambling with other phenomena such as anaphor binding and weak crossover, things are not as simple as they appear and the straightforward solution breaks down.

An altogether different approach is taken in this paper. The goal is to produce an analysis of scrambling that works for different languages and a wider variety of examples *without* introducing new machinery. The essential idea is to rely on the *same* (universal) constraints and parsing algorithms already in place for non-scrambling languages, e.g. English. In other words, we adopt the null hypothesis. So, we begin with a computationally-modelled linguistic framework that is already capable of handling scrambling as the deductive result of interactions of basic principles, such as general movement (Move-α) with Binding theory. The point is that scrambling (like the so-called "passive" and "dative" constructions) obeys the same restrictions already shown to be operative for other syntactic phenomena, and so should follow from in-

*The author is deeply grateful to Robert C. Berwick for his technical advice and comments.

[1] (Saito, 1992) remarks that *koto* 'the fact that' is often added to avoid the unnaturalness resulting from not having a topic in the matrix clause.

[2] It was brought to my attention by Doug Jones (personal communication), that German is normally considered to have only short distance scrambling for technical reasons. We will not explore this here, but note that none of the examples presented in (Becker et al., 1990) involve "scrambling" out of tensed clauses.

dependently justified principles; this is why it should be easy to add. Hence we gain, almost "for free", an account of its (rather subtle) interactions with previously described phenomena — not handled in the (Becker et al., 1990) account. As we will see, the system directly handles a surprisingly large number of examples from the recent literature.

However, as can be expected our experiments do reveal some surprises. The thoroughness of the parser in exploring all possibilities leads it to derive alternative analyses that are identical save for the presence of string vacuous scrambling. We note here that under more recent conceptions on movement e.g. (Chomsky, 1990), such options are never taken. Here, we will simply eliminate the unwanted alternatives without compromising empirical coverage by assuming that scrambling must be non-vacuous in the sense that every instance must be visible. We will translate this non-vacuity constraint into the LR(k)-parsing framework, and exhibit two different implementations, and end with a comparison of their computational efficiency.

But first, we consider a much wider variety of scrambling examples, including both positive and negative data (since scrambling in Japanese is not completely free), to show that simply adding permutation machinery to a base grammar cannot be descriptively adequate.

2 Interactions with Scrambling

Sentences (1)–(3) showed that Japanese exhibits non-clause-bounded, or long distance, scrambling as well as clause-bounded scrambling. Multiple scrambling is also possible; see (2b), (2c) and (3d). In each case, the *ni-* and *o*-marked objects have both been scrambled. Note that (3d), the long distance case, is classified by Saito as "unnatural but perfectly grammatical":

(3) (d) Bill$_j$-ni sono hon$_i$-o Mary-ga [$_{CP}$ John-ga t_i t_j watasita to] omotte iru (koto)

By drawing on (Saito, 1985),(1992) and (Tada, 1993), and by extension, on the extensive literature cited there, this section will summarize the data showing that Japanese scrambling is not only productive, but far from simple, exhibiting many subtle and complex interactions.

2.1 Basic Assumptions

It is not surprising that there are differences between the model described here and the theory assumed by Saito (1985). Originally, the system was designed to parse all and only the English sentences from "A Course in GB Syntax" (Lasnik & Uriagereka, 1988).[3] In subsequent work, see for example (Berwick & Fong, 1992), the system has been extended to handle basic examples in Japanese (from (Lasnik & Saito, 1984)) and Dutch. The basic modules in the current system are as follows:

- Move-α: with substitution and adjunction being the two basic operations and Subjacency. Also, that movement leaves traces.
- Binding theory, namely, Free Indexation along with Conditions A, B and C. Plus a simple PRO Control module.
- Case theory: structural and inherent Case, the Case filter and conditions on traces.
- Theta theory: including the Theta Criterion.
- Quantifier raising (QR) and *Wh*-movement at Logical Form (LF).
- The Empty Category Principle (ECP) operating at LF and S-structure.
- Elements of Full Interpretation at LF including licensing operator/variable constructions, reanalysis of $\bar{A}$-bound pro-forms, quantifier scoping, *Wh*-Comp condition from (Lasnik & Saito, 1984), and the like.
- Functional Determination for empty NPs. We make use of the following classes: variables, anaphors, pro and PRO, traces and empty operators.[4]

In all, there are about thirty principles. We assume basic phrase structure is binary branching and generated in accordance with $\bar{X}$-theory and the Projection principle. Furthermore, we currently assume only two functional categories, I and C, no VP-internal subjects, and finally that Japanese has SPEC(CP) only for LF movement and empty operators at S-structure (to handle relative clauses). Figure 1 shows a typical example of Japanese phrase structure as produced by the parser.

For scrambling, we will add two assumptions:

1. It is movement by adjunction in syntax; adjoining to either VP (short-distance) or IP (medium or long), and
2. The landing site is (tentatively) in an A-position.

Part of the evidence for assumption (1) will come, of course, from the data below; in other words, scrambling obeys the same kinds of constraints as for regular movement. As for the reasons for VP and IP, arguments are given in (Saito, 1985). Assumption (2) which will be revised later differs from (Saito, 1985), where it is assumed that scrambling is $\bar{A}$-movement. Despite this difference, it is surprising to see how many of Saito's examples actually go through. We note here that the A/$\bar{A}$-distinction is a crucial one since so many principles, e.g. Binding conditions, A-bound pro-form

[3] For a detailed description of the theory and implementation, see (Fong, 1991).

[4] Obviously, space limits us to a brief listing of the principles. However, note that this by no means a fixed nor exhaustive list.

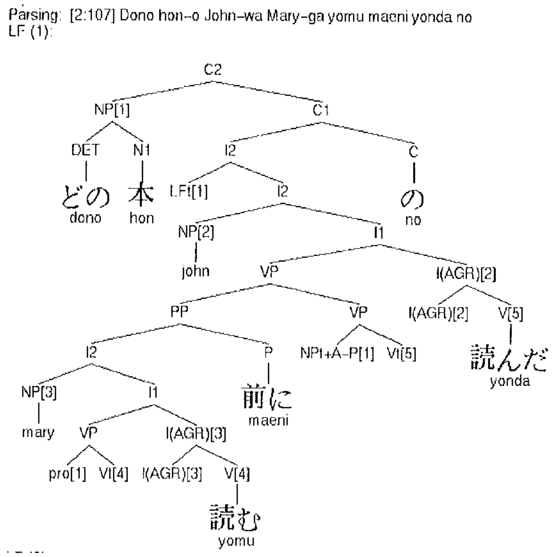

Figure 1: An Example of Japanese Phrase Structure

reanalysis and the ECP, and therefore analyses, turn on it. Much has also been written on this issue as it relates to scrambling in other languages, see (Mahajan, 1990) on Hindi and (Webelhuth, 1989) on Germanic languages.

2.2 Scrambling and Binding

Our goal in this and the following sections is to reinterpret the data in terms of the implemented theory outlined above, and see how far we get. We will point out any oddities and problems along the way. All examples are taken from (Saito, 1985) and have been verified to work on the parser unless otherwise stated.

(4) (a)* Kare$_i$-ga Mary-ga John$_i$-ni okutta tegami-o mada yonde inai (koto)[5]

(b) Mary-ga John$_i$-ni okutta tegami-o kare$_i$-ga mada yonde inai (koto)
The letter that Mary sent to John, he has not yet read it

(c)*? Masao-ga otagai$_i$-no sensei-ni karera$_i$-o syookaisita (koto)

(d) Karera$_i$-o Masao-ga otagai$_i$-no sensei-ni t_i syookaisita (koto)
Them$_i$, Masao introduced t_i to each other$_i$'s teachers

In each case, scrambling the pronoun or anaphor saves the ungrammatical sentence. (4a) is straightforwardly ruled out by Condition C of the Binding theory since *kare* binds *John*. (4c) and (4d) from (Saito, 1992) are also handled by Binding theory. In (4c) *otagai* is unbound which violates Condition A (all anaphors have to be A-bound). In (4d) it is A-bound since we have assumed scrambling to A-positions. Hence, these data involving the scrambling of anaphors and pronouns constitutes evidence that scrambled elements obey the usual principles.

2.3 Scrambling and Weak Crossover

Weak Crossover (WCO) effects are a well-known phenomenon in English. For example:

(5) (a) Who$_i$ loves his$_i$ mother

(b)* Who$_i$ does his$_i$ mother love t_i

(5a) can be interpreted as asking the question *for which person x, x loves x's mother*. Crucially, (5b) cannot be interpreted as asking the question: *for which person x, x's mother loves x*. In the parser, the unavailability of (5b) is explained by an LF principle, *Reanalyze Pro-Forms*, which reanalyzes as *variables* those pro-forms (e.g. pronouns) bound by an *operator* in an $\bar{A}$-position, as suggested by the above logical analysis. However, this fails when the pro-form intervenes between the operator and its trace, as is the case in (5b).

(6) (a) John$_i$-o kare$_i$-no hahaoya-ga t_i aisiteru
John$_i$, his$_i$ mother loves

(b) John-wa Mary-ga *pro*$_i$ yomu maeni sono hon$_i$-o yonda
John read that book$_i$ before Mary read it$_i$

(c)* John-wa Mary-ga *pro*$_i$ yomu maeni dono hon$_i$-o yonda no
Which book$_i$ did John read before Mary read it$_i$

(d) Dono hon$_i$-o John-wa Mary-ga *pro*$_i$ yomu maeni yonda no

(e)*? Soitu$_i$-no hahaoya-ga dare$_i$-o aisiteru no
Who$_i$ does 'the guy$_i$'s' mother love

(f) ? Dare$_i$-o soitu$_i$-no hahaoya-ga t_i-o aisiteru no

(g)* Karera$_i$-o Masao-ga otagai$_i$-no sensei-ni Hanako-ga t_i hihansita to itta (koto)
Them$_i$, Masao said to each other$_i$'s teachers that Hanako criticized t_i

(h)* Soitu$_i$-o hitome mita hito-ga Mary-ga dare$_i$-o sukininaru to omotta no
The person who took a glance at the guy$_i$ thought that Mary would fall in love with who$_i$

(i) * Dare$_i$-o soitu$_i$-o hitome mita hito-ga Mary-ga t_i sukininaru to omotta no

We note the following:

[5] As is conventional, subscript indices will be used to mark coreference possibilities. Roughly speaking, if two NPs have identical indices, then are said to denote the same object/person.

- In (6a), *John* "crosses over" *kare*. However, since *John* scrambles to an A-position by definition, the parser correctly reports no WCO violation. In Saito's original $\bar{A}$-based account, this example is problematic. [6]

- (6b) and (6c) show WCO effects can be found even with empty pronouns (*pro*). The parser rules out (6c) since *dono hon*$_i$ must raise at LF to the matrix clause.[7] No WCO violation is reported for the scrambled counterpart (6d). This is compatible with the A-position hypothesis. Running the parser produces the LF structure in figure 1.

- (6e) from (Saito, 1992) is the Japanese counterpart of the English WCO violation (5b). As expected, it is ungrammatical. On the $\bar{A}$-hypothesis, (6f) would be predicted to be as bad as (6e).

- (6g) and (6i) are both examples of long distance scrambling from (Saito, 1992) and (Tada, 1993)). We need to assume that long distance scrambling is to $\bar{A}$-positions to account for this under WCO, as in (Tada, 1993). We retain the A-position option for short distance scrambling only.[8] This is currently implemented as a stipulation. Note, empirical support for this dichotomy comes from Hindi, see (Mahajan, 1990).

Scrambling, by its interactive nature, also reveals shortcomings in the implemented theory. We now turn our attention to data not handled in the parser from (Saito, 1992):

(7) (a)? Dono hon$_i$-o Masao-ga Hanako-ga t_i tosyokan-kara karidasita ka siritagatteiru koto
The fact that which book$_i$, Masao wants to know Hanako checked out t_i from the library

(b) Zibunzisin$_i$-o Hanako$_i$-ga t_i hihansita (koto)

Herself$_i$, Hanako$_i$ criticized

Our essentially "for free" approach breaks down here. So far we have been successfully relying on existing principles to see us through the maze of scrambling facts. As Saito observes, ECP blocks the LF-lowering of *dono hon-o* in (7a).[9] However, in contrast to typical cases of ECP violations, Saito classifies (7a) as only being mildly ungrammatical. Similarly, *zibunzisin* A-binds *Hanako* in the (grammatical) example (7b). However, the parser reports a Condition C violation. According to Saito, the former case can be handled by making traces optional, and the latter by applying some form of LR Reconstruction.[10] We note that neither proposal is generally considered to be scrambling-specific and therefore points to the general incompleteness of the implemented system.

2.4 Scrambling and NQ Float

As a final case, consider the phenomenon of Numeral Quantifier (NQ) float, as shown in (8). Saito analyzed (8d) as an instance of scrambling, i.e. *sake* has been scrambled out to IP.

(8) (a) Sannin-no gakusei-ga sake-o nonde iru
3 students are drinking sake

(b) Gakusei-ga sannin sake-o nonde iru

(c) * Gakusei-ga sake-o sannin nonde iru

(d) Sake$_i$-o John-ga sanbon t_i motte kita
John brought 3 bottles of sake

Leaving aside the structure of NQ-NP, it is not clear whether *gakusei* in (8b) undergoes scrambling. Since Saito assumed that subjects do not scramble for inherent Case reasons — thereby explaining the ungrammaticality of (8c), it appears not to be the case.[11]

Finally, we observe that there are other cases we have not tested, such as clausal and adjunct scrambling, the effects of Subjacency, and the distinction between medium and short distance scrambling.

3 Scrambling and Computation

Although Japanese scrambling is complex, we have seen that by leveraging existing principles, many examples of short and long distance scrambling can be accommodated almost without change to the existing theory. At first glance, the same seems to be the case for computation. General phrase structure including the additional adjunction to IP and VP is covered by the existing LR(1)-based bottom-up shift-reduce parser (Knuth, 1965). The relation between a scrambled object and its launch site is computed as one part of the general rule of movement, Move-α.[12]

[6] An interesting point is that the similar sentence:

*John$_i$-o kare$_i$-ga t_i syookaisita (koto)

cited as an example of a crossover violation is traced to other reasons in the framework of the parser. It reports a Condition B violation irrespective of the A/$\bar{A}$-status of *John*. The trace t_i functions as PRO since it is locally A-bound by (*kare*) with an independent θ-role. Since the trace is an argument, it will violate one of the Binding Conditions.

[7] Under our assumptions, it undergoes LF *Wh*-movement to SPEC(CP), an $\bar{A}$-position, to rendezvous with *no*, the [+wh] element in HEAD(CP).

[8] Note this is not the only possible analysis. For example, Shin Watanabe (LSA, 1994) argues for scrambling as $\bar{A}$-movement only.

[9] Two points: (1) Saito refers to the Proper Binding Condition rather than the ECP. (2) *dono hon-o* lowers for the same reason it raises in (6e). See note 7.

[10] We note here that the interaction between LF Reconstruction effects and scrambling is also discussed in (Tada, 1993).

[11] In the implementation, NQ adjoins to NP and both NQ-NP and NP-NQ orders are made available.

[12] More precisely, the relation is *recovered* by a rule of Chain Formation. See (Fong, 1991) for the details of this and other mechanisms.

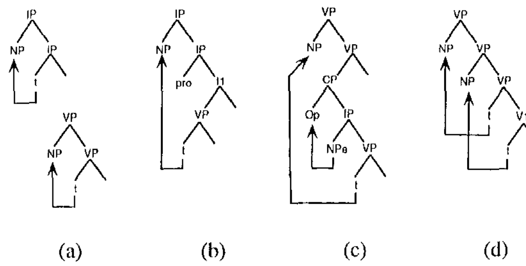

Figure 2: Examples of vacuous scrambling

The problem that arises is that the parser produces extra parses not eliminated by the existing principles, involving *vacuous scrambling* of the form shown in figure 2.

Let us define the obvious notion of "non-vacuous" or *visible* scrambling as scrambling that "passes over" one or more overt elements at S-structure. Adopting this definition immediately eliminates cases (a)-(c), and more importantly, the repeated application of case (a) which causes parsing non-termination.[13] In particular, this rules out vacuous scrambling over empty subjects, e.g. pro or PRO. As far the data from (Saito, 1985),(1992) and (Tada, 1993) goes, this is no restriction at all. This leaves case (d) which is vacuous only when considered as a "pair", i.e. each sub-instance is clearly non-vacuous.

We will make use of the following assumptions:

1. Scrambling is always to the left.
2. Empty NPs don't scramble.

Again, judging from the data, these appear to be no restriction all. Now, it is simple to implement the non-vacuity (NV) constraint as a licensing condition on phrasal adjunction:

$$\mathrm{IP}_i \rightarrow \mathrm{sNP},\ \mathrm{IP}_i(x), \quad \{\mathrm{NV}(x)\}$$
$$\mathrm{VP}_i \rightarrow \mathrm{sNP},\ \mathrm{VP}_i(x), \quad \{\mathrm{NV}(x)\}$$

Here, we assume that there are two phrasal adjunction rules, for IP and VP, that introduce scrambled NPs (sNPs).[14] Here, $\{\mathrm{NV}(x)\}$ is a *semantic action* which checks the frontier of x, the IP or VP phrase, for non-vacuity using the following left-to-right, two state machine:

State 0: (Start) See an overt node, go to State 1.
State 1: (End) Skip until see an NP gap, halt.

Note this is potentially inefficient since the NV constraint is only checked when the LR-machine completes the RHS of a rule, i.e. when it completes an IP or VP phrase. By operating at the level of the terminal string, instead of waiting for the complete IP or VP, we can take advantage of the fact that scrambling is always to the left to implement the non-vacuity check in a strictly left-to-right fashion. As before, when we see a potentially scrambled NP, we start the two state machine. Here, the basic idea is that a *shift* action (read a terminal) corresponds to the state 0 to state 1 transition. Similarly, a NP$\rightarrow \lambda$ *reduce* action corresponds to the "see an NP gap and halt" part. Comparing the two algorithms on 50 Japanese sentences (25 scrambling examples) from (Saito, 1985) & (Lasnik & Saito, 1984), an average speed-up factor of 2.3 per sentence and a total of 3.2 over the 50 sentences was observed for the second algorithm over just the phrase structure recovery portion. Due to the varying effects from the participation of other principles, the improvement for the total parsing time was less clear-cut, generally varying between no speed-up at all to a factor of two depending on the sentence. Using the 25 *non*-scrambling examples from the test set, an additional experiment between two variants of the parser, one with and one without the ability to handle scrambling, showed that the ability to handle scrambling exacts a penalty of about 30-50% in total parsing time. In conclusion, given the perhaps disproportionate effect of scrambling on parsing time, we suggest that although scrambling comes virtually for free linguistically speaking, the same certainly does not appear to be the case for computation.

[13]Note that the string vacuous empty operator movement shown in (c) does not count as an instance of scrambling. It's not adjunction at VP or IP.

[14]The tricky case of (d) shown earlier can be handled by restricting sNP to overt NPs only.

References

T. Becker, A.K. Joshi, & O. Rambow. 1990. Long-distance scrambling and tree adjoining grammars. In *COLING-90*.

R.C. Berwick & S. Fong. 1992. Madama butterfly redux: Parsing english and japanese with a principles-and-parameters approach. In R. Mazuka, editor, *Japanese Sentence Processing*. Lawrence Erbaum.

N.A. Chomsky. 1986. *Knowledge of Language: Its Nature, Origin, and Use*. Prager.

N.A. Chomsky. 1990. Some notes on economy of derivation and representation. In I. Laka & A. Mahajan, editors, *Functional Heads and Clause Structure*. MITWPL.

S. Fong. 1991. *Computational Properties of Principle-Based Grammarical Theories*. Ph.D. thesis, Artificial Intelligence Laboratory, MIT.

D. E. Knuth. 1965. On the translation of languages from left to right. *Information and Control*, 8(6):607–639.

H. Lasnik & M. Saito. 1984. On the nature of proper government. *Linguistic Inquiry*, 15(2).

H. Lasnik & J. Uriagereka. 1988. *A Course in GB Syntax: Lectures on Binding and Empty Categories*. MIT Press.

A. Mahajan. 1990. *The A/A-Bar Distinction and Movement Theory*. Ph.D. thesis, MIT.

M. Saito. 1985. *Some Asymmetries in Japanese and Their Theoretical Implications*. Ph.D. thesis, MIT.

M. Saito. 1992. Long distance scrambling in japanese. *Journal of East Asian Linguistics*, 1(1):69–118.

H. Tada. 1993. *A/A-Bar Partition in Derivation*. Ph.D. thesis, MIT.

G. Webelhuth. 1989. *Syntactic Saturation Phenomena and the Modern Germanic Languages*. Ph.D. thesis, University of Massachusetts.