An incremental approach to gapping in Japanese

Petter Haugereid
Western Norway University of Applied Sciences

Proceedings of the 26th International Conference on Head-Driven Phrase Structure Grammar
University of Bucharest
Stefan Müller, Petya Osenova (Editors)
2019
CSLI Publications
pages 42–57
http://cslipublications.stanford.edu/HPSG/2019

Keywords: HPSG, Gapping, Japanese, Incremental Parsing

Abstract

Gapping in Japanese, which is an SOV language, differs from gapping in SVO languages in that the conjuncts with the elided verbs appear in non-final position. In this paper I present an incremental approach to gapping in Japanese, where it is assumed that an argument structure type is constructed in the non-final clause(s) in the gapping construction. This type is unified with the construction type created by the final clause resulting in identical construction types for all conjuncts in the construction.

1 Introduction

Gapping is a phenomenon that poses a challenge to lexicalist approaches given the fact that the main verb of one or more of the conjuncts in these constructions is elided. Example (1) (from Sag et al. (1985)) shows the prototypical gapping construction with a transitive sentence in the first conjunct, and two arguments, but no verb, in the second conjunct.

(1) Kim likes Sandy, and Lee Leslie.

The constituents of the conjunct with the elided verb, Lee and Leslie, are referred to as the remnants, and the constituents that have their roles in the conjunct with the verb, Kim and Sandy, are referred to as correlates.

The examples in (2)–(6) demonstrate gapping in Japanese (from Kato (2006, p. 1–14)). (2) is a conjunction of two transitive sentences where the verb of the first conjunct is elided. (3) shows that the elided verb cannot be in the second conjunct, as in English. (4) shows that gapping also may occur with intransitive verbs. (5) shows that there may be four dependents in each conjunct, and (6) shows that there may be more than one conjunct with a gap.


(3) *John-ga hon-o katta sosite Mary-ga hana-o. John-NOM book-ACC bought and Mary-NOM flower-ACC

(4) John-ga kayobi-ni sosite Mary-ga doyoubi-ni hasiru. John-NOM Tuesday-ON and Mary-NOM Saturday-ON run ‘John runs on Tuesdays, and Mary on Saturdays.’


† I would like to thank three anonymous reviewers and the audience at the HPSG 2019 conference in Bucharest, Romania, for very useful comments and suggestions.
‘John bought books for Fred yesterday, and Mary flowers for Susan today.’

(6) John-ga hon-o sosite Mary-ga hana-o sosite Fred-ga pen-o sosite Sue-ga kitte-o katta.

‘John bought books, Mary flowers, Fred pens, and Sue stamps.’

According to Ross (1970), gapping operates forward in SVO languages like English. This is referred to as forward gapping (see (7)). And in SOV languages like Japanese, the verb appears in the last conjunct in gapping constructions (Ross, 1970). This is referred to as backward gapping (see (8)).

(7) a. SVO + SVO + SVO + ... + SVO ⇒
   b. SVO + SO + SO + ... + SO

(8) a. SOV + SOV + SOV + ... + SOV ⇒
   b. SO + SO + SO + ... + SO

Gapping in Japanese is sometimes equaled to Right Node Raising (Kato, 2006, p. 55). Yatabe and Tanigawa (2018) claim that Japanese does not have gapping, only Right Node Raising. They base their argument on the fact that the apparent ellipsis only is at the right node of the conjunct, illustrated in (9), where it appears that the whole right node nani o kau to yakusoku shita no is gapped. According to Yatabe and Tanigawa (2018), the reading of (10), where the verb kau and the complementizer to are not gapped, should be the same as the reading of (9) if Japanese had gapping, but this reading is not available, and they present this as evidence that Japanese does not have gapping.

(9) [Masao wa] ashita, (soshite) [Hanako wa] asatte
     [Masao TOP] tomorrow (and) [Hanako TOP] day after tomorrow
     [nani o] kau to yakusoku shita no?
     [what ACC] buy-PRES COMP promise do-PAST NML

   ‘What has Masao promised to buy tomorrow, and what has Hanako promised to buy the day after tomorrow?’

(10) ?* [Masao wa] ashita kau to, (soshite) [Hanako wa]
     [Masao TOP] tomorrow buy-PRES COMP (and) [Hanako TOP]
     asatte [nani o] kau to yakusoku shita
day after tomorrow [what ACC] buy-PRES COMP promise do-PAST
     no?
     NML

   ‘Same as (9)’
In this paper, I will not discuss whether or not gapping exists in Japanese. The aim will be to present an analysis of the examples in (2)–(6), where a verb is shared by two (or more) conjuncts. However, in the following I will refer to this phenomenon as gapping.

Gapping is a widely discussed phenomenon in the linguistic literature, and it is one of the hardest phenomena to handle in a grammar implementation. The analyses of gapping rarely find their way into grammar implementations. In this paper, the focus will be on implementability of accounts of gapping, and hence the perspective will be different from other, more theoretical, approaches. The hope is that it can complement the other approaches and show a way forward to how analyses of gapping can be implemented. Guided by limitations imposed by concerns about implementability and parser efficiency, the account I will present is limited in scope, and only accounts for a fraction of the data on gapping found in the literature.

2 Gapping in HPSG

In lexicalist theories, the syntactic structure is built up around heads which carry detailed information about the structure that will be built around them. This makes gapping constructions hard to account for, given that the verb, which is the head of the sentence, is missing.

Most HPSG approaches to gapping makes use of the linearization approach (Kathol, 1995; Beavers and Sag, 2004; Chaves, 2005; Crysmann, 2008; Kim and Cho, 2012). In this approach, the feature DOM(ain) (Reape, 1994) represents the linear order of phonological items, and this order is allowed to be different from the order in the constituent tree. This separation of linear order and constituent tree is powerful, and although relational constraints may be added to the grammar in order to impose restrictions on the order of the phonological items, it may put a heavy burden on the parser if it is not properly constrained.

Abeillé et al. (2014) present an alternative, construction-based HPSG approach to gapping. It is based on Mouret (2006), and does not make use of linearization. Instead it assumes that the constituents in the conjuncts with the elided verb, the remnants, form a non-headed constituent where the synsems of the remnants are entered onto a CLUSTER list in HEAD. This constituent undergoes a unary rule head-fragment-ph. This head-fragment rule checks the HEAD values of the remnants (via the CLUSTER list) against the HEAD values of the correlates, which the rule accesses via the context SAL(ient)-(sub)UTT(erase) feature (SAL-UTT) (see (12)).

1The analysis presented is possible to implement with the LKB system (Copestake, 2002).

2More complex examples of gapping, for example including chains of control verbs as in (11) (from Sag et al. (1985)) and examples like (9), will be topic for future research.

(11) Pat wanted to try to go to Berne, and Chris to Rome.
Syntactic constraints on head-fragment-ph (Abeillé et al., 2014)

\[
\text{head-fragment-ph} \Rightarrow \left[ \begin{array}{c|c|c}
\text{CONTEXT} & \text{SAL-UTT} & \text{CLUSTER} \\
\hline
\text{HEAD} & \text{MAJOR} & \text{HEAD} \\
\text{MAJOR} & + & \text{MAJOR} & + \\
\end{array} \right]
\]

In example (1), repeated here as (13), the correlates are Kim and Sandy. Consequently, their synsems can be accessed via the SAL-UTT feature. The head-fragment rule checks that their head values match with those of the remnants, Lee and Leslie. In this way, the subcategorization frames of the conjuncts with elided verbs are guaranteed to correspond to the subcategorization frames of the conjunct with the verb.

(13) Kim likes Sandy, and Lee Leslie.

The tree in Figure 1 is an illustration of how the features SAL-UTT and CLUSTER account for the matching of the argument frames of the initial conjunct and a conjunct with an elided verb.\(^3\)

In addition to the syntactic constraint shown in (12), there is a separate constraint on the head-fragment rule that assigns the semantic predicate that was assigned to the correlates, to the remnants. In (13), this means that the semantics of the second conjunct is like '(lee',leslie').

There are some challenges to the approach to gapping in Abeillé et al. (2014), and in particular the syntactic constraints on head-fragment-ph shown in (12). While it is possible to match the HEAD values of the synsems on the CLUSTER list with those on the SAL-UTT list, one needs to know the length of the SAL-UTT and CLUSTER lists, unless one introduces some extra functionality for list matching. If there are two correlates and two remnants, as in (13), one needs a head-fragment-ph type that has SAL-UTT and CLUSTER lists of length two, and which matches the HEAD values of the two first items and the HEAD values of the two second items. If there are three correlates and three remnants, one needs another head-fragment-ph type with lists of length three, and so on. In addition, if one allows the matching elements on the lists to come in different order, the number of matching rules required becomes large.

A more serious problem with the head-fragment rule is how to make the items on the SAL-UTT and CLUSTER lists accessible to the rule at the same time. The access to the correlates on the SAL-UTT list in the head-fragment rule presupposes that the conjunct with the correlates has been parsed when the head-fragment rule is applied, and that the correlates has been put on a SAL-UTT list. This list will have to be made accessible to the coordination rule, which pushes it down into the head-fragment rule, via the head-comps rule, as shown in Figure 1. The fact

\(^3\)The analysis presented in Abeillé et al. (2014) is more elaborate than the illustration shown here. It includes a functionality that allows them to match constituents with differing HEAD values, like adv and prep, and noun and adj.
that the CLUSTER list comes from below, and the SAL-UTT list comes from above, means that only one of the lists will be populated when the parser attempts to apply the head-fragment rule, irrespective of whether the parsing strategy is bottom-up or top-down. One of the lists will be empty until the whole coordination is parsed. This will lead to a large number of contexts where the head-fragment rules would be applicable, before both the lists are populated and the matching of the lists can be attempted, and it will lead to a massive burden of the parser.

Both the syntactic and semantic constraints assumed on the head-fragment rule in Abeillé et al. (2014) assume access to information in the conjunct with the verb. This makes sense in languages with forward gapping, like English, French and Romanian, but in a language like Japanese, where the remnants come before the correlates, one would have to wait for the final verb before the constraints required by the head-fragment rule would be made available. If one assumes a parser that works right-to-left, this can be accounted for, but it would be hard to defend from a psycholinguistic point of view.

In this paper, the incremental left-to-right approach to gapping in Haugereid (2017) will be adapted, and it will be shown how the left-to-right approach used to account for forward gapping also can be used for backward gapping, even though the verb only appears in the final conjunct. This is made possible given that the grammar is designed in such a way that a clause in principle can be parsed without
a verb. The argument structure is assumed to originate from the syntactic rules, and the verb is treated as a kind of obligatory modifier. If there is no verb, the parse will result in an underspecified construction type which only reflects the argument structure of the clause, but not the predicate of the main verb.

3 Analysis of gapping in Japanese

In Haugereid (2017), gapping in Norwegian, which is an SVO language, is accounted for by assuming that the predicate type of the first conjunct in a gapping construction is unified with predicates introduced by unary rules representing the elided verbs in the non-initial conjuncts. The predicate type reflects the argument structure of the clause, so the conjuncts with gapped verbs will have to realize the same type of arguments (for example a subject and an object) as the initial clause.

3.1 Incremental parsing and constituent structure

The constituent tree of the transitive sentence in (14) is assumed to be the flat structure in Figure 2a. The constituent structure is derived from the AVM of the parse tree, shown in Figure 2b. The step from parse tree to constituent tree involves the use of a feature STACK (Haugereid and Morey, 2012). In the following, the trees that will be presented, are parse trees, but they all have corresponding constituent trees.

(14) John-ga hon-o katta.
    John-NOM book-ACC bought
    John bought books.

Figure 2: Illustration of constituent tree and parse tree of a transitive sentence in Japanese

The parse starts in the bottom left corner with a START symbol. This symbol has the features shown in (15). In the illustration, there are three valence features
that all have negative values \( \text{arg1}–, \text{arg2}–, \text{arg3}– \). In addition, the value of the feature VBL is \( \text{synsem} \), which means that it requires a verb.

(15) \[
\begin{array}{c}
\text{START} \\
\text{VAL} \\
\text{VBL} \\
\end{array}
\begin{array}{c}
\text{CMP1} | \text{LINK} \text{arg1}– \\
\text{CMP2} | \text{LINK} \text{arg2}– \\
\text{CMP3} | \text{LINK} \text{arg3}– \\
\end{array}
\text{synsem}
\]

The \( \text{START} \) symbol is combined with the subject \( \text{John-ga} \) and the direct object \( \text{hon-o} \). There are separate valence rules for each of these functions, and they switch a negative link type to a positive. The rule for the direct object is shown in (16). It changes the negative link value \( \text{arg2}– \) in the first daughter to a positive link value in the mother \( \text{arg2}+ \).

(16) \[
\begin{array}{c}
\text{cmp2-struc} \\
\text{VAL} \\
\text{VBL} \\
\text{ARGS} \\
\end{array}
\begin{array}{c}
\text{CMP1} \\
\text{CMP2} \quad \text{LINK} \text{arg2}+ \\
\text{CMP3} \\
\end{array}
\text{synsem}
\]

At the top of the tree in Figure 2b, the verb is realized. This is done by the verb rule shown in (17). The rule takes as its first daughter a structure that requires a verb, and as its second daughter a verb, and it produces a structure that has saturated the verb requirement (VBL \text{anti-synsem}). In addition, the rule unifies all the link types with the \text{PRED} type of the verb.

(17) \[
\begin{array}{c}
\text{verb-struc} \\
\text{VAL} \\
\text{VBL} \\
\text{ARGS} \\
\end{array}
\begin{array}{c}
\text{CMP1} | \text{LINK} \\
\text{CMP2} | \text{LINK} \\
\text{CMP3} | \text{LINK} \\
\end{array}
\text{anti-synsem}
\]
The lexical entry for the verb ka (‘buy’) is shown in (18). It only has an ORTH value, a HEAD value, and a PRED value. There are no VAL features or ARG-ST list.

\[(18) \begin{array}{l}
\text{verb-lxm} \\
\text{ORTH} \langle \text{ka} \rangle \\
\text{HEAD} \text{verb} \\
\text{LKEYS} | \text{KEYREL} | \text{PRED} \text{ buy_prd}
\end{array} \]

Instead of the regular valence requirements associated with verb lexical items, the verb is given a PRED value \text{buy_prd}, and it is the position of this type in a type hierarchy of subconstruction types, that determines which argument frames that are possible for the verb. A simplified type hierarchy involving the type \text{buy_prd} is shown in Figure 3.

![Type hierarchy of subconstruction types, argument frame types, and construction types](image)

Figure 3: Type hierarchy of subconstruction types, argument frame types, and construction types

The hierarchy shows that the \text{buy_prd} type is compatible with two argument frames, a transitive frame \text{arg12_rel}, and a ditransitive frame \text{arg123_rel}. When the predicate is unified with one of these two frames, we get the construction types \_buy_12_rel and \_buy_123_rel, respectively. In this way, it is the type hierarchy of subconstruction types that determines which frames that are possible for a verb to enter.

The tree in Figure 4 shows how the linking types are changed from negative in the \text{START} node to positive in the top of the tree, and how the link types are unified with the PRED value of the verb. Since the types \text{arg1+}, \text{arg2+}, \text{arg3–}, and \text{buy_prd} are compatible (given the type hierarchy in Figure 3) the sentence is ultimately given a parse.

### 3.2 Analysis of gapping

SOV clause structure and backward gapping as demonstrated in (2)-(6) pose a challenge to the incremental left-to-right approach in Haugereid (2017). However, the
constructional approach allows for the construction of an argument frame type that is underspecified with regard to the predicate of the main verb of the clause. This is shown in Figure 3, where the type \( \text{arg12}_{\text{rel}} \) is the result of the unification of three subconstruction types: \( \text{arg1}^+ \), which is contributed by the rule that realizes the subject, \( \text{arg2}^+ \), which is contributed by the rule that realizes the direct object, and \( \text{arg3}^- \), which shows that no indirect object has been realized. (If an indirect object is realized, the \( \text{arg3}^- \) type will be replaced by \( \text{arg3}^+ \), resulting in the argument frame type \( \text{arg123}_{\text{rel}} \).) The tree in Figure 5 illustrates how the subconstruction types accumulate as the conjuncts in (2) are parsed.

In order to make the representation compact, I have used sets to illustrate the accumulation of the subconstruction types in Figure 5. In reality, each subconstruction type is the value of a separate feature. The underlining of subconstruction types in the tree represents the unification of these types.

---

Figure 4: Parse tree for Japanese transitive sentence
no more arguments are attached, a unary gapping rule (see (19)) unifies the LINK values, here \( \text{arg1+}, \text{arg2+} \) and \( \text{arg3–} \), with the PRED value of the KEYREL. (In the tree, this is marked by underlining the subconstruction types, and the elided verb is marked with the symbol \( \emptyset \).) The KEYREL value is unified with the GAPREL value. The rule switches the VBL value from \( \text{synsem} \) in the daughter to \( \text{anti-synsem} \) in the mother. It also gets the HEAD feature \( \text{GAPPING +} \).

(19)

\[
\begin{align*}
\text{verb-gapping-struc} & \\
\text{HEAD} & \left[\text{GAPPING +}\right] \\
\text{VAL} & \left[\begin{array}{c}
\text{CMP1} | \text{LINK} \\
\text{CMP2} | \text{LINK} \\
\text{CMP3} | \text{LINK}
\end{array}\right] \\
\text{VBL} & \text{anti-synsem} \\
\text{LKEYS} & \left[\begin{array}{c}
\text{KEYREL} | [\text{PRED}]
\text{GAPREL} | \emptyset
\end{array}\right] \\
\text{ARGS} & \left[\begin{array}{c}
\text{VAL} | \emptyset
\text{VBL} | \text{synsem}
\end{array}\right]
\end{align*}
\]

At this point, the three subconstruction types are unified, resulting in the argument frame type \( \text{arg12\_rel} \) (see the type hierarchy in Figure 3). The conjunct \textit{sosite}
initiates a new conjunct (see (20)), and it carries into the new clause the argument frame type from the gapping rule (see (21)).

(20) \[
\begin{align*}
&[\text{conj-word}] \\
&\text{ORTH} \langle \text{sosite} \rangle \\
&\text{HEAD} \begin{cases}
\text{conj} \\
\text{GAPPING} +
\end{cases}
\end{align*}
\]

(21) \[
\begin{align*}
&[\text{coord-struc}] \\
&\text{VAL} \begin{cases}
\text{CMP1|LINK arg1–} \\
\text{CMP2|LINK arg2–} \\
\text{CMP3|LINK arg3–}
\end{cases} \\
&\text{VBL} \begin{cases}
\text{synsem}
\end{cases} \\
&\text{LKEYS} \begin{cases}
\text{KEYREL} [\underline{\cdot}] \\
\text{ARGS} \begin{cases}
\text{HEAD} \begin{cases}
\text{GAPPING} [\underline{\cdot}]
\end{cases} \\
\text{VBL} \begin{cases}
\text{anti-synsem}
\end{cases} \\
\text{GAPREL} [\underline{\cdot}]
\end{cases}
\end{cases}
\begin{cases}
\text{coord-word} \\
\text{HEAD|GAPPING} [\underline{\cdot}]
\end{cases}
\end{align*}
\]

The second clause is parsed in the same manner, and at the top of the tree, the rule that attaches the verb, unifies the predicate \textit{buy.prd} with the subconstruction types of the second conjunct \((\text{arg1}+ , \text{arg2}+ , \text{arg3}–)\), resulting in the predicate type \textit{buy12.rel}. (The unified subconstruction types are underlined at the top of the tree in Figure 5). The rule also unifies this predicate type with the argument frame type carried over from the first conjunct \((\text{arg12.rel})\). In this way, the identity of the two construction types is ensured, and the two clauses get the same predicate.

The MRS (Copestake et al., 2005) for example (2), repeated below as (22) is given in Figure 6. The first \textit{buy12.rel} predicate is the result of unifying the construction type of the first conjunct \textit{arg12.rel} with the construction type of the last conjunct \textit{buy12.rel}.

(22) \text{John-ga hon-o sosite Mary-ga hana-o katta.}

\text{John-NOM book-ACC and Mary-NOM flower-ACC bought}

‘John bought books, and Mary flowers.’

The incremental subconstructional approach assumed in this paper is similar to the approach in Abeillé et al. (2014) in that the argument frame of the conjunct with the verb is unified with the argument frame of the conjuncts with the elided verbs. However, it differs from Abeillé et al. (2014), as well as other lexicalist approaches, in several respects. Firstly, this account is an incremental account. It differentiates between a parse tree (which is left branching, as shown in Figure...
Figure 6: Semantic representation – Gapping
5) and a constituent tree, which is relatively standard (see Haugereid and Morey (2012)). Secondly, this approach assumes a hierarchy of subconstruction types (Haugereid, 2009), as illustrated in Figure 3. It is this hierarchy of subconstruction types that accounts for the argument frames in the grammar, not the constraints on the lexical items, and this makes it possible to parse a sentence without a verb, as illustrated in the first conjunct in Figure 5. In standard HPSG, including Abeillé et al. (2014), verbs are specified with an ARG-ST, and there is no generalization over ARG-ST lists that corresponds to the hierarchy of subconstruction types assumed in my approach. A third difference between the approach in this paper and Abeillé et al. (2014) is the semantics. In my approach, the construction type that results from the unification of the subconstruction types, becomes the predicate of the verb (and the elided verb). The semantics is in this way integrated with the syntax. In Abeillé et al. (2014) however, there are separate constraints accounting for the syntax and semantics of gapping.

4 Future work

The suggested method accounts for the data in (2)–(6). There will be some overgeneration with regard to adjuncts, since they are not reflected in the argument structure of the verb. One solution to that would be to let not only information about arguments, but also adjuncts be carried over to the next conjunct. This is a topic for further investigation. Another foreseeable problem with the approach is the fact that the verb does not appear in the first conjunct. This will increase the search space of the parser, although it will be constrained by the hierarchy of subconstruction types. The search space could be further restricted if the method were to be combined with some kind of statistical ”guesser” for each word that is added.

References


