Hisao Tokizaki (ed.), Sapporo University.

Some Casual Notes on Grammar Comparison in Syntax: The Case of Analysis of Head-Final Structure*

Tomohiro Fujii
Yokohama National University

ABSTRACT. The paper concerns the question of how linguists can select the best grammar out of competing grammars. By working on an exercise problem from Japanese, it is shown that choosing among competing grammars can be harder than we hope even in this simple setting. A possible evaluation measure, which compares grammars in terms of their assumptions' simplicity, is proposed.

Keywords: grammar evaluation, head final structure, evaluation measure, simplicity

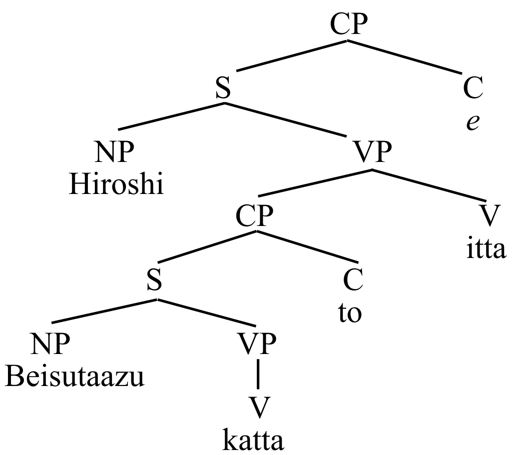
1. Introduction

(1) illustrates a type of Japanese complex sentence, which is often analyzed as containing an embedded clause headed by the complementizer *to* 'that'.

(1) Hiroshi-wa Beisutaazu-ga katta-to itta.
H-TOP BayStars-NOM won-COMP said
'Hiroshi said that the BayStars won.'

The standard structural analysis of this sentence type assumes what we can call the *S+C structure*, shown below.

(2)

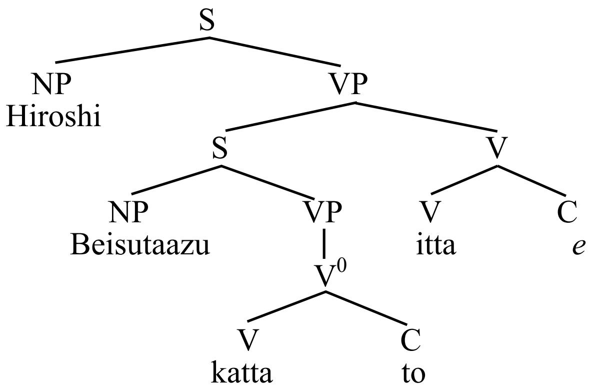


* The work was supported by JSPS KAKENHI Grant 16K02621.

In this analysis, the complementizer *to* is combined with an S to form a larger constituent, CP. (Throughout this paper, I use S rather than TP, which allows us to remain neutral as to how Tense participates in the structure.) It can be said that the S+C analysis has been widely accepted from the early 1970s on; see Nakau 1973, Inoue 1976, Shibatani 1978, among others.

Despite the wide popularity of the S+C structure, another view can be found in the literature. According to the view, the complementizer and the verb form a unit at surface structure, as in (3). We call this type of structure the *V+C structure* to distinguish it from the S+C structure introduced above (Sells 1995, Shimada 2007; see also Kitagawa 1986, Saito 2012). In the tree shown in (3), *katta* 'bought' and *to* 'that' form a constituent.

(3)



This sort of analysis looks attractive partly because it appears to capture the prosodic structure of the sentence type directly. That is, a tensed verb followed by a complementizer like *katta to* here is a prosodic unit.

In Fujii (2016), I conducted standard constituency tests to assess the structures of the construction exemplified by (1) and showed that the string consisting of the embedded subject and the verb, i.e. *Beisutaazu-ga katta* in (1), behaves as a constituent. I also showed that no data in favor of constituency of *katta to* are obtained by running such tests. Such results, as I claimed, should straightforwardly lead us to accept (2) and reject (3) as a syntactic representation of the construction at the relevant level of representation. As will be clear below, however, the decision seems to get subtler than we would hope when we talk in terms of grammar, not structure. Differently put, things are less transparent when we try to decide that a grammar is not good than when we decide that a structure is not good. Is that a problem for linguists? I believe it is. One standard approach to Universal Grammar (UG) has since Chomsky (1965) been by inferring what UG is like from the available information about the *grammar* of L_i and the Primary Linguistic Data of L_i. On the assumption that it is grammars that linguists test (and the child has to choose among), it would be worrisome if selection of right grammar(s) were too hard.

In what follows, we set up an exercise problem and work on it. In the present setting, we do not take into consideration any data points or arguments in favor of constituency of the embedded verb and the complementizer, unlike some of the works cited above. So, I assume that a grammar generating the S+C structure should be chosen as a better grammar for the specific data set over its counterpart generating the V+C structure. The problem to tackle is then how we can demonstrate the former being more highly valued than the latter. As will be clear later, when things are looked at from a grammar choice perspective, things can easily get subtle and it can be somewhat surprisingly difficult to rule out a grammar that assigns the V+C structure to string (1). Why does that happen? One reason is, I claim, that grammar selection may be hard partly because we reply too much on the outputs of competing grammars. I finally propose an evaluation measure that helps to see which of two grammars predicts what they predict in a *simpler* manner.

The next section reviews some data gathered by applying various diagnostic tests to the target sentence, including those reported in Fujii (2016).

2. Diagnostic Test Results

To find out how the complement construction is structured, I use the formulation of constituency test given in (4) and the definition of constituent given in (5).

(4) *Constituency Test*

If the string $m_1\ m_2\ ...\ m_n$ is moved, deleted or replaced by a pro-form, the terminals corresponding to $m_1\ m_2\ ...\ m_n$ form a constituent.

(5) *Constituent*

The terminals $m_1\ m_2\ ...\ m_n$, any of which can be null, form a constituent if and only if there is a non-terminal that dominates $m_1\ m_2\ ...m_n$ and no other terminals.

In addition, I refer to the string in (1) as $\mathbf{n_1\hat{}n_2\hat{}v_1\hat{}c\hat{}v_2}$ for expository purposes. The string *Beisutaazu-ga katta-to* then can be referred to as $\mathbf{n_2\hat{}v_1\hat{}c}$. We abstract away from case and topic marking entirely.

The data in (6), taken at a face value, suggest that $\mathbf{n_2\hat{}v_1\hat{}c}$ is a constituent.

(6) a. ***Movement test for*** $n_2\hat{}v_1\hat{}c$

Beisutaazu-ga katta-to Hiroshi-wa __ itta.
BayStars-NOM bought-COMP H-TOP said
'That the BayStars won, Hiroshi said __.'

b. ***Deletion test for*** $n_2\hat{}v_1\hat{}c$

Mari-wa Beisutaazu-ga katta-to iwanakatta-ga, Hiroshi-wa ∅ itta.
Mari-TOP BayStars-NOM bought-COMP said.not-but H-TOP said
'Mari didn't say that the BayStars won, but Hiroshi said ∅.'

c. ***Proform replacement test for*** $n_2\hat{}v_1\hat{}c$

Mari-wa Beisutaazu-ga katta-to iwanakatta-ga, Hiroshi-wa soo itta.
Mari-TOP BayStars-NOM bought- COMP said.not-but H-TOP so said
'Mari didn't say that the BayStars won, but Hiroshi said so.'

In (6a-c), $\mathbf{n_2\hat{}v_1\hat{}c}$ appears to have undergone movement, deletion and proform replacement. Note however that it is not so clear that these examples satisfy the antecedent of the statement in (4). For example, how do we know that the string $\mathbf{n_2\hat{}v_1\hat{}c}$ is moved as a group in (6a)? Purported "sub-movement" analyses like those shown in (7) are conceivable. In (7a), for example, the string in (6a) is analyzed as involving movement of *Beisutaazu-ga*, movement of *katta* and movement of *to*.

(7) ***Conceivable analyses of the movement example***

a. Beisutaazu-ga$_i$ katta$_j$ to$_k$ Hiroshi-wa __$_i$ __$_j$ __$_k$ itta.
b. Beisutaazu-ga$_i$ katta to$_j$ Hiroshi-wa__$_i$ __$_j$ itta.
c. Beisutaazu-ga katta$_i$ to$_j$ Hiroshi-wa__$_i$ __$_j$ itta.

The less evidence against these analyses we can collect, the weaker the evidence based on (6a) will be as evidence for constituency of $\mathbf{n_2\hat{}v_1\hat{}c}$. Fortunately, it is not too difficult to show that these sub-movements are implausible.

As shown in (8), all the possible sub-strings of $\mathbf{n_2\hat{}v_1\hat{}c}$ except *Beistaazu-ga* resist movement.

(8) ***Evidence against sub-movement analyses***

a. ?Beisutaazu-ga$_i$ Hiroshi-wa __$_i$ katta-to itta. (movement of $\mathbf{n_2}$)
b. *katta$_i$ Hiroshi-wa Beisutaazu-ga __$_i$ to itta. (movement of $\mathbf{v_1}$)
c. *to$_i$ Hiroshi-wa Beisutaazu-ga katta __$_i$ itta. (movement of **c**)

d. *katta-to$_i$ Hiroshi-wa Beisutaazu-ga __$_i$ itta. (movement of $\mathbf{v_1}$^$\mathbf{c}$)

e. *Beisutaazu-ga katta$_i$ Hiroshi-wa __$_i$ to itta. (movement of $\mathbf{n_2}$^$\mathbf{v_1}$)

One might still say that the true generalization could be that a sub-string (e.g. $\mathbf{v_1}$^$\mathbf{c}$) can move only when the rest (e.g. $\mathbf{n_2}$) moves. Such a possibility would have to be taken seriously if an example like (9) were grammatical; in (9), $\mathbf{n_2}$^$\mathbf{v_1}$ and **c** appear before the matrix subject in different order than they appear in (6a). The ungrammaticality of (9) calls into question this further analytic possibility. These considerations lead to the conclusion that (6a) is derived thought moving of $\mathbf{n_2}$^$\mathbf{v_1}$^$\mathbf{c}$ as a whole.

(9) ***Further evidence against sub-movement***

*katta-to$_j$ Beisutaazu-ga$_i$ Hiroshi-wa __$_i$ __$_j$ itta.
won-COMP BayStars-NOM H-top said

A similar concern applies to the deletion and proform examples. For a sub-deletion analysis like (10a) and a sub-replacement analysis like (10b) cannot be excluded a priori. Also, it is reasonable to worry about these possibilities since Japanese allows massive null arguments, indicated by $\varnothing_i$ in (10). If we found no evidence against these analyses, the judgements given in (6b-c) could not be used to argue for constituency of $\mathbf{n_2}$^$\mathbf{v_1}$^$\mathbf{c}$ confidently enough.

(10) ***Purported sub-deletion and sub-replacement analyses***

a. Mari-wa Beisutaazu-ga$_i$ katta-to iwanakatta-ga, Hiroshi-wa $\varnothing_i$ $\varnothing$ itta.
M-TOP B-NOM won-COMP said.not-but H-TOP said
'Mari didn't say that the BayStars won, but Hiroshi said so.'

b. Mari-wa Beisutaazu-ga$_i$ katta-to iwanakatta-ga, Hiroshi-wa $\varnothing_i$ soo itta.
M-TOP B-NOM won-COMP said.not-but H-TOP so said
'Mari didn't say that the BayStars won, but Hiroshi said so.'

In (10a), $\mathbf{n_2}$ and $\mathbf{v_1}$^$\mathbf{c}$ are allegedly deleted independently. Similarly, in (10b), the proform *soo* allegedly replaces $\mathbf{v_1}$^$\mathbf{c}$. The available data, however, fail to support these analyses. (11a) and (11b) show that where deletion and replacement apply to $\mathbf{v_1}$^$\mathbf{c}$ unambiguously, unacceptable sentences ensue. (I assume that **c** cannot be null in embedded finite clauses in Standard Japanese.)

(11) ***Evidence against sub-deletion and sub-replacement***

a. Mari-wa Doragonzu-ga katta-to itta-ga,
M-TOP D-NOM won-comp said-but
*Hiroshi-wa Beisutaazu-ga ∅ itta.
H-TOP B-NOM said
'Mari said that the Dragons won, but Hiroshi said that the BayStars won.'

b. Mari-wa Doragonzu-ga katta-to itta-ga,
M-TOP D-NOM won-COMP said-but
*Hiroshi-wa Beisutaazu-ga soo itta.
H-TOP B-NOM so said
'Mari said that the Dragons won, but Hiroshi said the BayStars won.'

These data cast doubt on the idea that (6b-c) can be derived through deletion/replacement of **v_1^c**.

So the data in (6)-(11) suggest two things: that the string **n_2^v_1^c** is a constituent and that the string **v_1^c** never passes the constituency tests.

Now let's proceed to test constituency of the string **n_2^v_1**. Curiously enough, **n_2^v_1** does not pass the movement test though it passes the other two tests. (12a) shows that moving **n_2^v_1** causes unacceptability. (12b) and (12c) involve complementizer-stranding deletion and 'indeterminate-proform replacement', respectively. The latter examples are acceptable.

(12) a. ***Movement test for n_2^v_1***
*Beisutaazu-ga katta Hiroshi-wa __ to itta.
BayStars-NOM won H-TOP COMP said
'That the BayStars won, Hiroshi said that __'

b. ***Deletion test for n_2^v_1***
A: Beisutaazu-wa katta-no?
BayStars-TOP won-Q
'Did the BayStars win?'
B: ∅ -tte Hiroshi-wa itta-yo.
-COMP H-TOP said-PART
'Hiroshi said that ∅.'

c. ***Proform replacement test for n_2^v_1***

A: Hiroshi-wa nan-te itta-no?
H-TOP what-COMP said-Q
'What did Hiroshi say?'
B: Hiroshi-wa Beisutaazu-ga katta-to itta-yo.
H-TOP BayStars-NOM won-COMP said-PART
'Hiroshi said that the BayStars won.'

As is the case with constituency of **n$_2^v_1$^c**, although **n$_2^v_1$** undergo constituency-sensitive processes without difficulty as in (12b-c), purported sub-deletion and sub-replacement analyses need to be excluded in order to establish constituency of the string.

Let's examine indeterminate *nani*-replacement, first. The question is whether we find evidence against analyzing *nani* in (12c) as a proform for **v$_1$** followed by a null NP referring to *Beisutaazu*, as in (13). Evidence against this analysis exists. (14) independently demonstrates that such *backward proform formation* is clearly unavailable.

(13) ***Purported sub-replacement analysis***
B: Hiroshi-wa $\varnothing_i$ nan-te itta-no?
H-TOP what-COMP said-Q
A: Hiroshi-wa Beisutaazu-ga$_i$ katta-to itta-yo.

(14) ***Evidence against v$_1$-replacement***
A: Hiroshi-to hanasi-ta-yo.
'I talked with Hiroshi.'
B: Hiroshi-wa $\varnothing_i$ katta-to itta-no?
H-TOP won-COMP said-Q
'Did he say $\varnothing_i$ won?'
A: #Hiroshi-wa Beisutaazu-ga$_i$ katta-to itta-yo.
H-TOP B-NOM won-COMP said-PART
'Hiroshi said that the BayStars won.'

We thus safely assume that in (12c), **n$_2^v_1$**, rather than **v$_1$**, is replaced by the indeterminate proform.

Next, is there any empirical reason to argue against the alternative sub-deletion analysis of (12b), where two deletion sites would be involved before the stranded complementizer, as in (15)? (16) below shows that the verb, unlike the subject NP, cannot be deleted in isolation. This initially looks like evidence against the sub-replacement analysis, but a confounding factor makes the unacceptability of (16b) little informative.

(15) ***Purported sub-deletion analysis***

A: Beisutaazu-wa$_i$ katta$_j$ -no?
'Did the BayStars win?'

B. $\varnothing_i$ $\varnothing_j$-tte Hiroshi-wa itta-yo.
-COMP Hiroshi-TOP said-PART

(16) ***Apparent evidence against sub-deletion***

a. B: $\varnothing_i$ katta-to Hiroshi-wa itta-yo.
won-COMP H-TOP said-PART

b. B. *Beisutaazu-wa $\varnothing_j$-tte Hiroshi-wa itta-yo.
BayStars-TOP -COMP H-TOP said-PART

Deletion leading to particle stranding is generally restricted to sentence-initial position. As in (17), particle stranding becomes impossible when pronounced material precedes the particle.

(17) ***Confounding factor for counter-argument to sub-deletion***

B: (*Hiroshi-wa) $\varnothing$ -tte itta-yo.
H-TOP -COMP said-PART

This means that we cannot use the unacceptability of (16b) to refute the hypothetical deletion of $\mathbf{v_1}$. We thus seem to have to conclude that the data do not contradict the alternative story appealing to deletion of $\mathbf{n_2}$ followed by deletion of $\mathbf{v_1}$.

To summarize, we have discussed the following nine main data points.

(18) ***Behavior of the string $n_2\hat{}v_1\hat{}c$***

d_1. $\mathbf{n_2\hat{}v_1\hat{}c}$ undergoes movement.

d_2. $\mathbf{n_2\hat{}v_1\hat{}c}$ undergoes deletion.

d_3. $\mathbf{n_2\hat{}v_1\hat{}c}$ undergoes proform replacement.

Behavior of the string $n_2\hat{}v_1$

d_4. $\mathbf{n_2\hat{}v_1}$ does not undergo movement.

d_5. It is not clear that $\mathbf{n_2\hat{}v_1}$ can be deleted.

d_6. $\mathbf{n_2\hat{}v_1}$ undergoes proform replacement.

Behavior of the string v_1^c

d_7. **v_1^c** does not undergo movement.

d_8. **v_1^c** does not undergo deletion.

d_9. **v_1^c** does not undergo proform replacement.

Though data point d_7 does not argue for any analysis over another as alluded to above, the other eight can be used to evaluate grammars and, hopefully, to choose the best grammar from them.

3. Which Grammar Fits Better with the Data?

We compare three grammars based on the data gathered in Section 2. Let's start with simple context-free grammars in (19) and (20), called G_1 and G_2, respectively. G_1 generates the S+C structure, while G_2 the V+C structure. (For the current purposes, we assume that a context-free grammar consists of a set of terminal symbols, a set of non-terminal symbols, a set of start symbols and a set of production rules; see Partee et al 1990, Carnie 2010.)

(19) **G_1**: Set of terminals ={Hiroshi, Beisutaazu, itta, katta, to, *e*},
Set of non-terminals ={CP, C, S, NP, N, VP, V}
Start symbol ={CP}
Set of production rules = {S→ NP VP, NP → N, VP → V, VP → CP V,
CP → S C,
N → Hiroshi, N → Beisutaazu, V → itta, V → katta, C → to, C → *e*}

(20) **G_2**: Set of terminals ={Hiroshi, Beisutaazu, itta, katta, to, *e*},
Set of non-terminals ={S, NP, N, VP, V^0, V, C, *e*}
Start symbol ={S}
Set of production rules = {S → NP VP, NP → N, VP → V^0, VP → S V^0,
V^0 → V C,
N → Hiroshi, N → Beisutaazu, V → itta, V → katta, C → to, C → *e*}

G_1 and G_2 are similar in several ways. First, G_1 and G_2 generate the same strings. They only produce two patterns of sentence. NNVCVC and NVC, and they have the same *lexical rules* (i.e. rules introducing terminal nodes: two terminals for N, two for V and two for C). So, either grammar generates 2×2×2×2×2×2 sentences of the NNVCVC pattern and 2×2×2 sentences of the NVC pattern. Secondly, as one might immediately notice from the first similarity, the grammars generate a lot of unacceptable sentences. For instance:

(21) *Beisutaazu-wa Hiroshi-ga katta *e* katta to.
B-TOP H-NOM bought bought COMP

Non-sentences like this arise because neither grammar incorporates a device such as subcategorization features. Third, both grammars lack a device to deal with case and topic marking.

Now let's see how G_1 differs from G_2 in the ability to cover the data set that we saw in Section 2. In a nutshell, G_1 fits with the data better than G_2 does. Before proceeding, note that neither grammar, as it stands, can derive sentences involving movement, deletion and proform replacement. As we will see soon below, however, while it is quite easy to extend G_1 so that these sentences can be covered under it, it is difficult to extend G_2 that way. Thus, a difference between the two grammars lies in ease of extension (See Larson 2010 for further illustrations with English data.)

To make G_1 handle the data, we add the rules shown in (22) to develop it. (We are agnostic about details including where a moving element moves to, what happens to the place that the element has moved from, how to choose among kinds of proforms, etc.) The structures in (23) are ones generated by G_1+.

(22) ***New rules***

a. Apply a transformation rule R to a single sub-tree T, whose top node is not zero-level, i.e. not immediately above a terminal node. R $\in$ {movement, deletion, proform replacement}.

b. C $\rightarrow$ no

(23) ***Structures generated by G_1+***

a. [$_{CP}$ [$_{CP}$ **n_2 v_1 c_1**] [$_S$ **n_1** ___ **v_2**] *e*]

b. [$_{CP}$ [$_S$ **n_1** [$_{CP}$ proform] **v_2**] *e*]

c. [$_{CP}$ [$_S$ **n_1** [$_{CP}$ $\varnothing$] **v_2**] *e*]

d. [$_{CP}$ [$_S$ **n_2 v_1**] [$_S$ **n_1** [$_{CP}$ ___ to] **v_2**] *e*]

e. [$_{CP}$ [$_S$ **n_1** [$_{CP}$ [$_S$ proform] **c**] **v_2**] no]

As (23a-c) suggest, the data points concerning movement, deletion and preform replacement of **n_2^v_1^c** are properly covered under G_1+.

As for the data points concerning **n_2^v_1**, G_1+ fails to capture its immobility (d_4) by mistakenly allowing **n_2^v_1** to move to sentence initial position, as shown in (23d). But the grammar covers d_6 correctly, predicting successful proform replacement of **n_2^v_1** as shown in (23e).

Finally, data points d_7-d_9, which concern immobility, unerasablity and unreplaceability of **v_1^c**, just follow from G_1+: **v_1^c** is never a constituent in this grammar. This way, G_1+ covers seven of the eight data points.

How easy is it to extend G_2 to cover the eight data points? G_2 assigns the structure below to the target sentence.

(24) ***Structure generated by G_2***

[$_S$ **n_1** [$_{VP}$ [$_S$ **n_2** [$_{VP}$ [$_{V0}$ **v_1 c**]] [$_{V0}$ **v_2** *e*]]]

We can make G_2 consistent with d_1-d_3 without difficulty. Just adding the same transformation rule as (22a) suffices, as noted in (25a-c) below. Regarding immobility of **n_2^v_1**, it directly follows from G_2+ since the string cannot be a constituent. By contrast, the fact that **n_2^v_1** can be substituted by a proform is problematic. The grammar incorrectly bars such proform replacement. Finally, the grammar fails to capture all the relevant behavior of **v_1^c**. The string is a constituent in G_2+ and therefore the grammar overgenerates for d_7-d_9. This is shown in (25d-f).

(25) ***Structures generated by G_2+***

a. [$_S$ [$_S$ **n_2** [$_{V0}$ **v_1 c_1**]] [$_S$ **n_1** ___ [$_{V0}$ **v_2** *e*]]

b. [$_S$ **n_1** [$_{CP}$ proform] [$_{V0}$ **v_2** *e*]]

c. [$_S$ **n_1** [$_{CP}$ ∅] [$_{V0}$ **v_2** *e*]]

d. [$_{V0}$ **v_1 c_1**] [$_S$ **n_1** [$_S$ **n_2** __ [$_{V0}$ **v_2** *e*]]

e. [$_S$ **n_1** [$_{VP}$ [$_S$ **n_2** [$_{VP}$ [$_{V0}$ proform]] [$_{V0}$ **v_2** *e*]]]

f. [$_S$ **n_1** [$_{VP}$ [$_S$ **n_2** [$_{VP}$ [$_{V0}$ ∅]] [$_{V0}$ **v_2** *e*]]]

The results are summarized in the table below. ("↑" and "↓" indicate that the grammar encounters overgeneration and undergeneration, respectively.)

(26) ***Grammars' outputs with respect to the main data points***

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
	Behavior of **n_2^v_1^c**			Behavior of **n_2^v_1**			Behavior of **v_1^c**		
G_1+				↑	NA				
G_2+					NA	↓	↑	↑	↑

The table clearly shows that G_2+ captures fewer data points than G_1+.

Before proceeding, two comments are in order. First, one might ask what if we formulate proform replacement and deletion as rewriting rules like "CP → ∅", "S → nani", as opposed to general transformation as described as (22). This move does not affect G_1 much but seems to help G_2 hide some of its weaknesses. That is, if we decide not to add any rules like "V^0 →∅" or "V^0 → soo", G_2+ stops giving wrong results for d_8 and d_9. I do not explore this issue any further, but it should be noted that G_1+ would still cover more data points than this revised version of G_2+.

The second comment has to do with overgeneration and undergeneration. Overgeneration problems are relatively easy to fix, compared to undergeneration problems. This is because one can in principle revise a grammar by adding a constraint to stop it from generating the ungrammatical sentences. G_1+'s inadequacy concerning the immobility of $\mathbf{n_2\hat{}v_1}$ can be remedied that way. We can revise the grammar by proposing that, say, movement to S to the edge of CP is prohibited because it is too short (Abels 2003). Given the 'solvable in principle' nature of overgeneration problems, we should choose G_1+ over G2+ not only because G_1+ accounts for more data points than G_2+, but also because G_1+ does not have a hard-to-fix undergeneration problem that G_2+ does. All this seems reasonable as ways of evaluating competing grammars based on their fit with empirical data.

In Section 4, we discuss another grammar that assigns the V+C structure to the target sentence.

4. A Third Grammar

Now, as one might have already noticed quickly, there is an immediate possibility we should consider. It is, so to speak, the "union" of G_1 and G_2. Call the third grammar G_3. In (23), the symbols and rules in bold face are those that are borrowed from G_1.

(27) **G_3**: Set of terminals ={Hiroshi, Beisutaazu, itta, katta, to, *e*}
Set of non-terminals ={**CP**, **S′**, NP, N, **VP′**, V, **S** ,**VP**, **V^0**, V, C}
Set of start symbols ={S, **CP**}
Set of production rules = {**S′ → NP VP′**, **NP → N**, **VP′ → V**,
VP′ → CP V, **CP → S′ C**,
S→ NP VP, VP → V^0, VP → S V^0, V^0 → V C,
N → Hiroshi, N → Beisutaazu, V → itta, V → katta, C → to, C → *e*}

Two things should be noted here. First, the new grammar is far from elegant, compared to G_1 (and perhaps G_2). It has more non-terminals, more productions rules and more start symbols than the previous grammars, although the set of strings it generates remains the same as the set of strings that G_1 (and G_2) generates. Second, G_3 can be understood as a version of G_2 revised with the goal of covering constituency of $\mathbf{n_2\hat{}v_1}$. G_3 allows derivations yielding a constituent consisting of $\mathbf{n_2\hat{}v_1}$, as well as those yielding a constituent consisting of $\mathbf{v_1\hat{}c}$. G_3 therefore has the property of making the string $\mathbf{n_1\hat{}n_2\hat{}v_1\hat{}c\hat{}v_2}$ structurally ambiguous between the two structures below. Note that this is a kind of structural ambiguity that do not seem to have any semantic effects.

(28) a. $[_{CP}\ [_{S'}\ \mathbf{n_1}\ [_{VP'}\ [_{CP}\ [_{S'}\ \mathbf{n_2}\ [_{VP'}\ \mathbf{v_1}]]\ \mathbf{c}]\ \mathbf{v_2}]]\ e]$

b. $[_{S}\ \mathbf{n_1}\ [_{VP}\ [_{S}\ \mathbf{n_2}\ [_{VP}\ [_{V0}\ \mathbf{v_1}\ \mathbf{c}]]\ [_{V0}\ \mathbf{v_2}\ e]]]$

G_3+, a version of G_3 armed with the movement rule and appropriate production rules, then, can handle some of the data points that G_2+ cannot. The table gets updated as in (29).

(29) ***Grammars' outputs with respect to the main data points***

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
	Behavior of $\mathbf{n_2\hat{}v_1\hat{}c}$			Behavior of $\mathbf{n_2\hat{}v_1}$			Behavior of $\mathbf{v_1\hat{}c}$		
G_1+				↑	NA				
G_2+					NA	↓	↑	↑	↑
G_3+				↑	NA		↑	↑	↑

Column d_4 shows that overgeneration with S-movement is carried over from G_1+ to G_3+, but, as shown under column d_6, the undergeneration problem found in G_2+ is now circumvented under G_3+. Columns d_7-d_9 show that adding G_1-like production rules of course do not help to block the V+C structure. At this point, it is largely clear that G_1+ is better than G_3+ in terms of the fit of their outputs with the data points.

Now suppose that one can successfully propose an excellent, non-ad hoc, independently motivated solution to the overgeneration problems with G_3+ for d_7-d_9. In that case, how should we go about grammar evaluation? One might say that we cannot reject G_3+ right away and might keep it as a candidate grammar of Japanese depending on how nice the solution to the problems is. Others might react that G_3+ should be ruled out because the G_2-originated production rules of G_3+ are unmotivated to begin with. The observation made so far suggests to me that potential difficulty in grammar choice arises here and the reason is because we rely too heavily on grammars' predictions for the empirical data points. To put it another way, it seems that we have been evaluating grammars too *extensionally* in the sense that we do not care much *how* the grammars make the predictions they make.

The current state of affairs reminds us of early generative grammar's efforts to build and develop evaluation measures (Chomsky 1957, 1964, 1965). An evaluation measure is meant to tell which of two grammars is valued more highly than the other even when a data set cannot distinguish them. The following passage is from Chomsky (1965:42).

> The problem is to devise a procedure that will assign a numerical measure of valuation to a grammar in terms of the degree of linguistically significant generalization that this grammar achieves. The obvious numerical measure to be applied to a grammar is length, in terms of number of symbols. But if this is to be a meaningful measure, it is necessary to devise notations and to restrict the form of rules in such a way that significant considerations of complexity and generality are converted into considerations of length, so that real generalizations shorten the grammar and spurious ones do not.

To evaluate grammar simplicity, we might count symbols as suggested in the passage above. We might also count rules. That is, a grammar with fewer rules is better than the other grammar. The idea of counting rules of grammars, however, does not seem easy to implement. The number of rules of a grammar changes after it is revised. Thus, even if a grammar G_i has fewer rules than another grammar G_j for a data set D_i, we do not confidently decide if that will be the case after a new data set D_j is presented to test the grammars. Larson (2010) shows a concrete toy example to illustrate it. The 'VP-less' grammar with S→N V and some lexical rules is simpler, by definition, than the 'VP' grammar with S→N VP, VP→V and the same lexical rules. But this evaluation of the two grammars does change once we attempt to accommodate sentences like *Mary ran and slept* and *Mary ran and slept and swam*. The former grammar needs as many new rules as it encounters new examples, whereas the latter can handle them with the minimum change, incorporating the recursive rule VP→VP Conj VP. Now the VP grammar is simpler than the VP-less grammar. This way, such uncertainty could make rule counting less useful as an evaluation measure.

5. Length of Assumption List

I would like to consider a possible simplicity measure, which utilizes what I call *Assumption Lists*. Here I aim to show that the metric allows us to choose G_1 over G_3 more confidently than when we focus on the extension or the outputs of each grammar. I cannot afford to discuss, though, whether the metric could be applied to a wider range of cases or to more sophisticated grammars like ones with multiple levels of representation.

It seems quite straightforward that G_1 accounts for the data set using four assumptions. The boundary conditions for the present discussion include (i) the eight data points we have seen — as seen above, d_5 is excluded since it is not informative — and (ii) the assumption that movement, deletion and proform replacement are captured as transformation rules affecting single non-minimal subtrees. Here is G_1's Assumption List for the eight data points, which has length 4.

(30) ***G_1+'s Assumption List for the eight data points***

i. **n_2^v_1^c** is a constituent. [Accounts for d_1-d_3]

ii. **n_2^v_1** is a constituent. [Tested against d_4 and d_6. Accounts for d_6 but not d_4]

iii. **n_2^v_1** fails to move for some reason. [Accounts for d_4]

iv. **v_1^c** is not a constituent. [Accounts for d_7-d_9]

As will be clearer in comparison with G_2 below, the nicest feature of G_1 is that it successfully *compresses* three data points (d_7-d_9) into one statement that **v_1^c** is not a constituent ((30iv)), rather than spending three assumptions to cover the three data points. This property of the grammar gives it a shorter Assumption List.

The Assumption List for G_3 has length 7 in total and length 4 for d_7-d_9.

(31) ***G_3+'s Assumption List for the eight data points***

i. **n_2^v_1^c** is a constituent. [Accounts for d_1-d_3]

ii. **n_2^v_1** is a constituent [Tested against d_4-d_6. Accounts for d_6 but not d_4]

iii. **n_2^v_1** fails to move for some reason. [Accounts for d_4]

iv. **v_1^c** is a constituent. [Tested against d_7-d_9]

v. **v_1^c** fails to move for some reason. [Accounts for d_7]

vi. **v_1^c** fails to be deleted for some reason. [Accounts for d_8]

vii. **v_1^c** fails to be replaced by a proform some reason. [Accounts for d_9]

The assumption that **v_1^c** is a constituent ((31iv)), put together with the fact **v_1^c** does not pass any constituency test, forces the system to add three more assumptions, (31v-vii), to make itself compatible with the data points. Thus the data points d_7-d_9 either contradict the grammar or make its Assumption List longer.

At this point, it should be stressed that G_3's List can be shortened by compressing (31v-vii) into one. [This is comparable to the situation toward the end of Section 4.] Namely, we may find out that **v_1^c** fails to move, get deleted or become a proform for the same, unified reason. But G_3 would still have length 5, spending two assumptions to take care of d_7-d_9. G_1 would be still simpler, and therefore better, than the sophisticated version of G_3.

Lastly, we quickly look at G_2's Assumption List.

(32) ***G_2+'s Assumption List for the eight data points***

i. **n_2^v_1^c** is a constituent. [Accounts for d_1-d_3]
ii. **n_2^v_1** is not a constituent [Tested against d_4-d_6. Accounts for d_4 but leads to a hard-to-fix wrong prediction for d_6.]
iii. **v_1^c** is a constituent. [Tested against d_7-d_9]
iv. **v_1^c** fails to move for some reason. [Accounts for d_7]
v. **v_1^c** fails to be deleted for some reason. [Accounts for d_8]
vi. **v_1^c** fails to be replaced by a proform some reason. [Accounts for d_9]

As mentioned earlier and noted in (32ii), the assumption that **n_2^v_1** is not a constituent causes G_2 undergeneration of one grammatical sentence pattern. To the extent that the problem is serious, G_2 is never chosen. Putting aside this problem, though, it can be said that G_2 is simpler than G_3 in the simplicity terms adopted here, since the former's Assumption List is of length 6 while the latter's is of length 7.

6. Summary

By working on an exercise problem, I have argued that grammar comparison can be hard to work out if it is conducted too extensionally, i.e. by relying too much on the outputs of grammars under comparison. I have suggested that we need an evaluation measure of sorts to handle cases where competing grammars may look equally good when evaluated in terms of their outputs. In our exercise problem, we have examined a case in which the apparently quite bad grammar for the data set gets quite close, if not equal, in its predictive power to the grammar that we think is well motivated. The proposed measure using Assumption List Length successfully allows us to choose the former over the latter in a more assured manner.

References

Abels, Klaus. 2012. *Phases: An essay on cyclicity in syntax*. Berlin: de Gruyter.

Carnie, Andrew. 2010. *Constituent structure*. New York: Oxford University Press.

Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1964. *Current issues in linguistic theory*. The Hague: Mouton.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Fujii, Tomohiro. 2016. Fukubun no kozo to umekmi-hobun no bunrui. *Nihongo bunpo handobukku*, ed. by Keiko Murasugi, Mamoru Saito, Yoichi Miyamoto and Kensuke Takita, 2-37. Tokyo: Kaitakusha.

Inoue, Kazuko. 1976. *Seisei bunpo to nihongo*, Tokyo: Taishukan.

Kitagawa, Yoshihisa. 1986. *Subject in Japanese and English*. Doctoral dissertation, University of Massachusetts, Amherst.

Larson, Richard. 2010. *Grammar as science*. Cambridge, MA: MIT Press.

Nakau, Minoru. 1973. *Sentential complementation in Japanese*. Tokyo: Kaitakusha.

Partee, Barbara H., Alice ter Meulen and Robert E. Wall. 1993. *Mathematical methods in linguistics*. Dordrecht: Kluwer.

Saito, Mamoru. 2012. Case checking/valuation in Japanese. *Nanzan Linguistics* 8: 109-127.

Sells, Peter. 1995. Korean and Japanese morphology from a lexical perspective. *Linguistic Inquiry* 26, 277-325.

Shibatani, Masayoshi. 1978. *Nihingo no Bunseki.* Tokyo: Taishukan.

Shimada, Junri. 2007. Head movement, binding theory and phrase structure, ms., MIT.