A Minimalist Parsing Account of Attachment Ambiguity in English and Korean

So Young Lee

Stony Brook University
soyoung.lee@stonybrook.edu

Abstract

This paper investigates whether attachment ambiguity preference attested in psycholinguistic literature can be correctly predicted, utilizing memory based metrics developed by Graf et al. (2017). Two different types of attachment ambiguity in Korean, the dative argument attachment ambiguity and the relative clause attachment ambiguity, are mainly tested. Furthermore, in order to examine whether the Minimalist Grammar Parsing Model can predict the broad human sentence processing effects, the work in the previous literature in Graf et al. (2017) were tested together. While none of the single metrics made the correct predictions, the combination of metrics based on tenure and size could predict the preferences of all different kinds of constructions reported in psycholinguistic studies. This suggests that human sentence processing effect cannot be understood by a single element. The well-selected metrics in this paper can tell us what kind of structural complexity are related to processing sentences.

Keywords: Minimalist Grammar Parsing Model, attachment ambiguity, computational approach, structural complexity

1. Introduction

In this paper, I will show that syntactic complexity is related to the attachment preference in English and Korean in computational linguistic view. I mainly focus on two different types of attachment ambiguity in
Korean: the dative argument attachment ambiguity and the relative clause attachment ambiguity, where the previous theoretical proposals, such as Gibson et al. (1996)’s Recency based on linear locality and Felser et al. (2003)’s Predicate proximity based on structural locality, fail to explain two phenomena uniformly.

One type of attachment ambiguity discussed in this paper is a dative attachment ambiguity observed in Korean. Due to the availability of scrambling and null pronouns in Korean, a dative argument such as *ttal-eykey* “to/for daughter” can have different interpretations when it appears at the earlier part of a sentence, as shown in (1). In the interpretation in (1a), *ttal-eykey* is located in the main clause of the sentence; in the interpretation in (1b), *ttal-eykey* is located in the relative clause.

(1) a. Main verb attachment

‘John showed his daughter the car that the uncle bought for him or her.’

\[\begin{align*}
\text{John} &-i \quad \text{ttal} &-eykey & \text{[samchon} &-i \quad \text{pro} &-ij \quad t_k \\
\text{John}_{\text{Nom}} & \quad \text{daughter}_{\text{Dat}} & \quad \text{uncle}_{\text{Nom}} & \quad \text{pro} & \quad t \\
\text{sacwu-n] & \quad \text{cha} &-k-lul & \quad \text{poyecuessta}. \\
\text{bought}_{\text{Rel}} & \quad \text{car}_{\text{Acc}} & \quad \text{showed}
\end{align*}\]

b. Relative clause verb attachment

‘John showed someone else from context the car that the uncle bought for his daughter.’

\[\begin{align*}
\text{John} &-i \quad [\text{ttal} &-eykey \quad \text{samchon} &-i \quad t_i \quad t_j \\
\text{John}_{\text{Nom}} & \quad \text{daughter}_{\text{Dat}} \quad \text{uncle}_{\text{Nom}} \quad t \quad t \\
\text{Sacwu-n] & \quad \text{cha} &-k-lul & \quad \text{poyecuessta}. \\
\text{bought}_{\text{Rel}} & \quad \text{car}_{\text{Acc}} & \quad \text{showed}
\end{align*}\]

Their English translations show that the two different structures give rise to separate meanings. It has been empirically shown in previous literature (Koh 1997, Kiaser 2007) that main verb attachment of dative arguments,
i.e., the reading in (1a), is the preferred one. This has been offered as an example to support the psycholinguistic theories based on linearity such as the garden path model (Frazier 1987) and recency (Gibson et al. 1996). When the dative argument is first encountered, the structure built at that point is only the matrix clause. Consequently, the matrix clause is the only place where the dative argument can be attached.

(2) Garden path theory and Recency: Attach new incoming materials to the most recently processed phrase if grammatically possible. (Frazier 1987, Gibson et al. 1996)

However, these theories cannot explain the preference shown in other attachment ambiguities in Korean such as relative clause attachment ambiguity.

The relative clause attachment ambiguity examined in this paper is in (3) and (4).

(3) English

Someone shot the maid of the actress [that was standing on the balcony].

(4) Korean

Nwukwun-ka [palkhoni-ey se-iss-nun] yepaywu2-uy
Someone Nom balcony-Loc stand-Prog-Rel actress-Gen
kacengpwu1-lul sswassta
maid Acc shot

‘Someone show the maid of the actress that was standing on the balcony.’

The relative clause that was standing on the balcony can be attached to NP1 maid or NP2 actress. Depending on which NP the relative clause is attached to, it can be either Low Attachment (LA) or High Attachment
(HA); LA refers to the attachment to the structurally lower NP and HA refers to the attachment to the structurally higher NP. Relevant experimental studies investigating the attachment preference of the relative clause in NP complex constructions such as (3) and (4) have shown cross-linguistic variations: LA is preferred in English (Cuetos and Mitchell 1988, Phillips and Gibson 1997) while HA is preferred in Korean (Jun 2003, Lee and Kweon 2004, Ha 2005).

LA preference in English can be successfully illustrated by the garden path model and recency. Between the two possible attachment places, the lower NP is the most recently processed phrase, and therefore the relative clause in English is attached there. Since these theories, however, consider the linear locality only, they also predict LA preference in Korean, which is in opposition to the empirical results (HA preference). In the same vein, given the preference of the dative attachment ambiguity in (1), the processed relative clause predicts that its head will appear. When the lower NP is encountered, it is attached to the relative clause as its head, which seems to imply LA preference. For this reason, the Garden path model and recency fails to account for HA preference in Korean. Hence, other studies including Felser et al. (2003) attempt to explain HA preference in (4) in terms of structural locality.

(5) Predicate proximity: Attach as structurally close as possible to the head of a predicate phrase. (Felser et al. 2003)

However, the predicate proximity cannot explain LA preference in English. As a result, the psycholinguistic theories in the previous literature fail to capture the phenomena in attachment ambiguity in a uniformed way as in (6).
Therefore, this paper aims to show that the phenomena in attachment ambiguity can be uniformly explained by MG Parsing Model (Stabler 1996, 2013) based on syntactic complexity. The advantages of MG parsing are that it incorporates ideas from theoretical syntax related to syntax processing and it allows us to precisely quantify the memory usage related to processing sentences. With reference to the results of previous psycholinguistic studies on attachment ambiguity, I will demonstrate that the MG parser in particular is successful in predicting the observed preference between two equally viable constructions (1), (3), and (4), and explore the reasons behind it. Furthermore, by investigating the wide range of constructions, we will be able to contribute to understanding the mechanisms of human sentence processing. Thus, this paper will examine the results reported in Graf et al. (2017) that discussed the relative clause processing effect (Subject Relative Clause (SRC) vs. Object Relative Clause (ORC)) with the attachment ambiguity at the end.

This paper is structured as follows. An overview of main approaches related to relative clauses, the way MG parser works, and the memory based metrics are discussed in Section 2. The psycholinguistic data on dative attachment ambiguity in Korean and their evaluation are given in section 3. The psycholinguistic data on relative clause attachment ambiguity in English and Korean and their evaluation appear in section 4. Additionally, I will discuss the results by including the data on the relative clause processing effect reported in the previous literature in section 5. Section 6 is the conclusion of this paper.

<table>
<thead>
<tr>
<th></th>
<th>Recency</th>
<th>Predicate proximity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dative argument: Matrix clause attachment preference</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Relative clause: Low attachment preference in English</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Relative clause: High attachment preference in Korean</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>
2. Background

2.1 Minimalist Grammars

In order to quantify the prediction of the processing effect on ambiguous sentences, Minimalist Grammars (MGs) in Stabler (1996, 2013) are adopted. In this section, I will briefly introduce MGs and a MG Parser.

MGs are a highly lexicalized formalization of the Chomskian Minimalist Program (1995). They consist of lexical items and derivational features. In MGs, the trees are built by means of two main derivational operations, Merge and Move, by calculating the encoded features on the lexical items. The features on lexical items are mainly categorized into four groups: selectional features (+F), categorical features (-F), licensor features (+f) and licensee features (-f). The selectional features and categorical features involve the Merge operation. The licensor and licensee features involve the Move operations. An example tree built up by MGs is as follows:

(7) a. Derivation tree

(Graf et al. 2017)
b. Phrase structure tree

(7a) is the derivation tree for the topicalized sentence *John, the girl likes*. (7b) is the phrase structure tree corresponding to (7a). As can be seen in (7a), all the information related to syntactic operations Merge and Move are specified on each lexical item and each derivation step is labeled on the internal nodes.

In MGs, feature calculation proceeds from the leftmost feature of the head to the right, and only one feature is involved in each derivational step. The type of derivational features such as selectional features or licensor features decides what kind of syntactic operation should be applied there. For the Merge operation, the selectional features are checked by merging with the lexical item carrying the corresponding categorical features. For instance, in (7a), the first Merge operation is applied to lexical items *likes* and *John*. This Merge operation is licensed by successfully checking the selectional feature +D on *likes* and the categorical feature -D on *John*. After remaining merging steps are done, Move operations take place. For instance, the unchecked feature +acc on V will trigger Move operation. The lexical item which has
\( -\text{acc} \) feature will be moved up to the spec of \( \text{v} \) and both feature \((+\text{acc} \text{ and } \text{V} \text{ and } -\text{acc \ on \ John}) \) will be checked. The remaining steps will be done in the same fashion. The result of feature calculation yields exactly the same word order in the phrase structure tree.

Now let us see how the MG parser works. If MGs are like the manual to implement the syntactic trees, the MG parser is like the instructions to compute the given syntactic trees. A top-down MG parser scans the leaf nodes in the derivation tree by following a particular order in a phrase structure tree. The algorithms of traversing the derivation tree are as follows (Kobele et al. 2013). Basically, a top-down MG parser read the nodes from top to bottom and from left to right. However, since the surface order of lexical items in the derivation tree such as (7a) is not the same as the surface order of lexical items (7b) derived from the phrase structure tree, simply scanning the leaf nodes of the derivation tree from left to right yields the wrong word order. Thus, the MG parser must keep tracking the derivation operations which affect the linear word order while scanning the nodes. The annotated version of the derivation tree of (7a) in (8) shows how the MG parser works.

(8) Annotated tree

(Graf et al. 2017)
In the tree, the superscripted number to the left of each node is an index, which indicates when the node is introduced into the memory stack. The subscripted number to the right side of each node is an outdex, which indicates when the node comes out from the memory stack. For instance, the scanned word order should be the same with the order of leaf nodes read from left to right in the phrase structure tree, so the first leaf node which must be scanned in (8) is *John*. In order to search *John*, the MG parser travels from the top to the bottom and from left to right. While traveling, all the intermediate nodes from the upper most node CP to *John* ‘s sister node *likes* are introduced into the memory. They stay in the memory until they can be discharged following the word order of the phrase structure tree. In this way, the parser will scan every terminal node in the derivation tree.

In the next section, the way to quantify the memory usage by calculating the index and the outdex will be discussed.

2.2 Parsing Metrics

The memory usage on processing can be measured by the metrics developed by Graf et al. (2017). They are utilized for psycholinguistic predictions in this paper. In this section, I will briefly introduce the metrics in Graf et al. (2017).

The memory based metrics (MAX, BOX, SUM, and so on) can quantify the memory usage in the three different ways: 1) Tenure (how long a node stays in the memory), 2) Payload (how many nodes are in the memory at the same time), 3) Size (How many bits the node consumes in memory). The definitions of these three notions are as follows.

(9) a. Tenure: In a derivation tree, for every node n with index and outdex, its tenure is the difference between its index and its outdex. A node’s tenure is trivial iff its tenure is equal to 2 or smaller than 2.

b. Payload: The payload of the derivation tree is the number of nodes with non-trivial tenure.
c. Size: For every node \( n \), its size is the number of phrases that are reflexively dominated by \( n \), distinct from \( n \), and are associated to a Move node that reflexively dominates \( n \).

The non-trivial nodes, whose tenure is greater than 2, is based in the derivation trees. Tenure and Payload in (9) are easily calculated by using index and outdex. First, Payload is simply counting the number of nodes with non-trivial tenure. The Metric of Payload is referred to as Box because the outdex of non-trivial nodes is boxed (Graf and Marcinek 2014).

\[
\text{(10) Box: } |\{n|\text{has tenure >2}\}|
\]

Next, the metrics measuring the memory usage based on the tenure are as follows (Graf et al. 2017).

\[
\text{(11) } T \text{ is the set of nodes of derivation tree } t
\]

\[
\text{a. MaxT: } \max(\{\text{ten}(n)|n \in T\})
\]

\[
\text{b. SumT: } \sum_{n \in T, \text{ten}(n)>2} \text{ten}(n)
\]

\[
\text{c. AvgT: } \text{SumT}(t)/\text{Box}(t)
\]

MaxT is indicating the maximum memory usage of a single node in the given derivation tree. SumT is the total tenure of non-trivial nodes in the tree. AvgT is the average memory usage of each non-trivial node. Let us consider the derivation tree (8) again. Max T in (8) is 10(T). SumT is 39 (= 8(C)+10(T)+5+8+8) which reports the total tenure of non-trivial nodes in that tree. AvgT is 7.8 (=39/5).

In addition to three metrics above, Graf et al. (2017) suggested the metrics measuring memory usage based on size. The nodes which go through Move operation are involved. They are named Movers, SumS, MaxS and AvgS.

\[
\text{(12) a. Movers: } |M|, \text{ where } M \text{ is the set of all nodes that are the root of a subtree undergoing movement.}
\]
b. SumS: $\sum_{m \in M} i(m) - f(m)$  
c. MaxS: $\max (\{i(n) - f(n) | n \in T\})$, where $T$ is the set of all nodes of the derivation tree  
d. AvgS: $\text{SumS}(t) \div \text{Movers}(t)$

In (12), $M$ is the set of all root nodes of a subtree undergoing movement. Movers is the number of the root of a subtree undergoing movement. For example, in (8), two phrases (*the girl* and *John*) go through the movement, so Movers is 2. SumS is basically measuring the length of filler and gap: $i(m)$ is the index of $m$ and $f(m)$ is the index of the node where $m$ is moved up. That is, SumS is the sum of the difference of the root nodes index and the landing sites index for each item going through movement. In (8), SumS is 12 (= $(9-1)+(7-3)$). MaxS is 8, which is the longest length between the filler and the gap (*John*). AvgS in (8) is 6 (= $12/2$). Graf and Marcinek (2014) suggest another metric MaxT^R which is a recursive application of MaxT. When two trees get the exactly same value for MaxT, it compares the next highest payloads of the two trees recursively until they find the difference between them. For example, the set of payloads in (8) is [10, 8, 8, 8, 5] and let us suppose that the set of payloads of another tree is [10, 8, 5]. Then, MaxT predicts the same difficulty in both trees but according to MaxT^R, (8) is predicted more difficult because of the third ranked payload 8 is greater than the third ranked payload 5 of the other tree. The same strategy is applied to MaxS^R.

There are more refined metrics by node types such as interior nodes, leaf nodes, pronounced nodes, and unpronounced leaf nodes.

(13) Each metric $M$ has four refined metrics by node type.

a. $M_1$: the metric $M$ is applied to only interior nodes  
b. $M_L$: the metric $M$ is applied to only leaf nodes  
c. $M_P$: the metric $M$ is applied to only pronounced leaf nodes  
d. $M_U$: the metric $M$ is applied to only unpronounced leaf nodes
Each metric mentioned in (10), (11), and (12) is applied to the specific targeting nodes.

At last, Box is simply counting non-trivial nodes, so it does not consider how many nodes are held in the memory at the same time. Graf et al. (2017), therefore, introduce two additional memory usage metrics, Con and Div.

\[(14)\]
\[
\text{a. Con(vergence): } |\{<u,v>|\text{ten}(u) \geq 2,\text{ten}(v) \geq 2, \text{index}(u) \leq \text{index}(v) \leq \text{outdex}(u)\}| \\
\text{b. Div(ergence): } |\{<u,v>|\text{ten}(u) \geq 2,\text{ten}(v) \geq 2, \text{outdex}(u) \leq \text{index}(v)\}| \\
\]

When the index and outdex of non-trivial nodes (u and v) are not overlapped, they are not in the memory simultaneously. For example, if the outdex of the node u is 5 and the index of the node v is 7, u is already discharged when v entered into the memory. The metric Con counts the pair of two non-trivial nodes which are in the memory at the same time but the metric Div counts how many pairs of nodes are not in the memory at the same time. In this paper, a total of 1600 metrics are used to evaluate memory usage by including ranked complexity metrics. The ranked complexity metric is a ranked pairing of the metrics mentioned above, such as <MaxT, SumT>. Similarly to ranked constraints in OT, MaxT is used first and SumT is used for the cases which are not correctly predicted by MaxT. I will show that the very limited number of metrics can precisely predict the processing effect on the attachment ambiguity. They will be a guideline to reveal what kind of factors caused the processing burden.

2.3 Relative Clauses

Before we jump into the evaluation of the constructions of attachment ambiguity, the main analysis of relative clauses will be discussed in this section because the syntactic constructions utilized in this paper center around relative clauses (RC).

There are two main approaches to the relation between the RC head and the RC internal gap: the Head External Analysis (Chomsky 1977, Montague 1974, Partee 1975) and the Head Raising Analysis (Brame 1968, Schachter
The former suggested that the RC head is externally generated out of RC as in (15).

\[(15)\]

```
  DP
   the NP
      bike CP
         C'
             C TP
                 T'
                     T vP
                         I v'
                             v VP
                                 saw which
```

Instead of the head, the RC operator is substituted in the subject or object position in the RC depending on the RC type and is then moved to the edge of RC. This is also named the wh-movement analysis.

Contrary to Head External Analysis, Head Raising Analysis postulates that the RC head is base-Generated inside the RC and goes through the movement to position of the head RC as in (16).
This type of Head Raising Analysis is called the promotion analysis. The promotion analysis has become dominant with regards to not only the RCs of head-initial languages such as English but also to those of head-final languages such as Japanese and Korean. Both of these analyses will be evaluated in this paper.

3. Testing Metrics with Dative Argument Attachment Ambiguity

3.1 The Syntactic construction and Psycholinguistic results

As shown in (17)\(^1\), four different sentences are tested in this paper. (17a) and (17b) have the same lexical items except the main verb. Due to the semantic meaning of the matrix verb, the dative argument daughter is preferably attached to a matrix verb in (17a), but it is attached to the verb of a relative clause in (17b) because the matrix verb is a transitive verb. Even though (17c) and (17d) are the same surface sentence, for the reason that their internal syntactic structures are different in terms of its meaning. They

\(^1\) (17a) and (17b) are shortened version of the data in Koh (1997), and (17c) and (17d) are shortened version of data in Kweon et al. (2004).
are distinguished here.

(17) IO-DO

a. Main verb attachment (Unambiguous matrix verb)
   \[Emma_Nom-ka \; ttal\text{-}eykey \; [samchon-i \; pro_i \; t_j \; sacun-n]\]
   \[Mom_{Nom} \; daughter_{Dat} \; uncle_{Nom} \; pro \; t \; bought_{Rel}\]
   \[cacenke\text{-}lul \; mulyeocuessta.\]
   \[bike_{Acc} \; handed \; down\]

   ‘A mom handed down her daughter the bike that uncle bought for her.’

b. Relative clause verb attachment (Unambiguous matrix verb)
   \[Emma_Nom-ka \; [ttal\text{-}eykey \; samchon-i \; t_i \; t_j \; sacun-n]\]
   \[Mom_{Nom} \; daughter_{Dat} \; uncle_{Nom} \; t \; t \; bought_{Rel}\]
   \[cacenke\text{-}lul \; cohahayssta.\]
   \[bike_{Acc} \; liked\]

   ‘A mom liked the bike that uncle bought for her daughter.’

c. Main verb attachment (Ambiguous matrix verb)
   \[Phikulleysi_{Nom}-i \; Lopinl\text{-}eykey \; [Phwu-ka \; pro_i \; t_j\]
   \[Piglet_{Nom} \; Robin_{Dat} \; Pooh_{Nom} \; pro \; t\]
   \[ttacun-n] \; kkwul\text{-}lul \; phallassta.\]
   \[picked_{Rel} \; honey_{Acc} \; sold\]

   ‘Piglet sold the honey that Pooh picked up for him to Robin.’

d. Relative clause verb attachment (Ambiguous matrix verb)
   \[Phikulleysi_{Nom}-i \; [Lopinl\text{-}eykey \; Phwu-ka \; t_i \; t_j\]
   \[Piglet_{Nom} \; Robin_{Dat} \; Pooh_{Nom} \; t \; t\]
   \[ttacun-n] \; kkwul\text{-}lul \; phallassta.\]
   \[picked_{Rel} \; honey_{Acc} \; sold\]

   ‘Piglet sold the honey that Pooh picked up for Robin (to somebody).’
There is no consensus on the base word order of Korean ditransitive structure yet, so both word orders (IO-DO and DO-IO) are considered. In addition, each sentence is analyzed by two different types of relative clause analysis, wh-movement analysis and promotion analysis. Hence, a total of 16 (=4x2x2) constructions are evaluated. The annotated derivation trees for (17a) and (17b) are shown in (18). The trees which have IO-DO base word order are only provided in (18).

(18) a. The tree for (17a)
Since MGs do not offer a headedness parameter to determine the linearization of arguments, this paper assumes that Korean right word order (SOV) is derived from SVO via movement. For the same reason,
prenominal relative clauses in Korean are also supposed to be derived from postnominal relative clause via movement (Yun et al. 2015). This assumption causes additional movements in Korean and it increases the value of memory usage metrics because they are very sensitive to the movement. Since the movement to get the right word order, however, is equally applied to every sentence in Korean, this does not affect the result when MGs predict the preference of attachment ambiguity based on the score of memory usage metrics.

According to an eye movement tracking study (Koh 1997) and self-paced reading study (Kiaser 2007), the main verb attachment is processed more easily than the relative clause verb attachment ((17a)>>(17b)). Even though there is no psycholinguistic literature to compare the processing difficulty of the totally ambiguous sentences such as (17c) and (17d), it is expected in the light of the results of Koh (1997) and Kiaser (2007)’s studies that (17c) would be processed faster than (17d). Based on these observations, the metrics will be evaluated in the next section.

3.2 Results

The values of 1600 metrics are compared\(^2\). The results are provided in (19) and (20). The tables in (19) and (20) include the only metrics that correctly predict the matrix verb attachment preference in all four different constructions. The Yes in each cell indicates the correct prediction. For each metric, pair wise comparisons of the derivation trees were conducted. For instance, the minimally paired trees such as (17a) and (17b) which have the same base word orders of ditransitive and the relation between the RC head and the gap are compared. The comparison results are separately provided in (19) and (20) depending on the relative clause analyses (promotion analysis and wh-movement analysis).

---

\(^2\) 1600 metrics cannot be compared manually so I used the python program which automatically calculates the values of metrics. This python program is available in the Github repository of the Stony Brook Computational Linguistics lab: https://github.com/CompLab-StonyBrook.
(19) The results (promotion analysis)

<table>
<thead>
<tr>
<th></th>
<th>IO-DO</th>
<th>DO-IO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unambiguous Matrix Verb</td>
<td>Ambiguous Matrix Verb</td>
</tr>
<tr>
<td>Avg T</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Avg T_{PU}</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Avg T_{U}</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Box T_{I}</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Box T_{IU}</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Sum T</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Sum T_{U}</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The basic metrics, Avg T and Sum T, and the refined metrics, Avg T_{PU}, Avg T_{U}, Box T_{I}, Box T_{IU}, and Sum T_{U} predict the preference of a matrix verb attachment to a relative clause verb attachment in all four different constructions in promotion analysis. The results of basic metric Box T are not shown in (19) because they were tied except the third listed construction (an unambiguous matrix verb and DO-IO base word order). Since the value of Box T indicates the number of non-trivial nodes, the results of the metric Box T shows that the total number of non-trivial nodes is same in compared constructions except the third one. In other words, the reason why the basic tenure metrics (Sum T and Avg T) can correctly predict the preference is that the values of non-trivial nodes in the less preferred constructions are heavier than the ones in the preferred ones.

Recalling that the tenure metrics are measuring how long the nodes stay in memory, the result tells us that the nodes in the relative clause verb attachment constructions stay longer in memory in some reason. Taking a look at how parser performs here in detail, we can see that it is caused by the additional movement (the scrambling of a dative argument to the left edge of the relative clause). Specifically, while the parser is searching the scrambled dative argument in relative clauses, all nodes which appear before the dative argument such as C, subject and T in a relative clause in the
derivation tree are held in the memory. Those nodes’ tenure increased. In other words, the additional movement raises the memory usage and this incurred the less preference for the relative clause verb attachment.

Next, the results of wh-movement analysis are presented in (20).

(20) The results (wh-movement analysis)

<table>
<thead>
<tr>
<th></th>
<th>IO-DO</th>
<th>DO-IO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unambiguous Matrix Verb</td>
<td>Ambiguous Matrix Verb</td>
</tr>
<tr>
<td></td>
<td>(17a) vs (17b)</td>
<td>(17c) vs (17d)</td>
</tr>
<tr>
<td>AvgT</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgT_U</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MaxS</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MaxS^R</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT_U</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The basic metrics AvgT, SumT, and MaxS and the refined metrics AvgT_U, MaxS^R, and SumT_U predict the preference of a matrix attachment ambiguity in all four different constructions at the same time. Similar to the results of the promotion analysis, the tenure based metrics mainly succeed at predicting the preference. With the same reason in the promotion analysis, the high tenure of the nodes in the relative clause verb attachment construction caused the less preference for them. These results also show that the structural complexity triggered by scrambling in the relative clause raises the memory usage.

In sum, the metrics mentioned above tell us that the processing difficulty in dative argument attachment ambiguity is caused by more memory usage in the structures including scrambling.
4. Testing Metrics with Relative Clause Attachment Ambiguity

4.1 The Syntactic construction and Psycholinguistic results

In this section, the ambiguity of the attachment of a relative clause to a complex noun phrase will be discussed. The result will clearly show that the structural complexity should be necessarily considered as the factor that affects sentence processing. The tested phrases are in (21) and the annotated trees of (21a) are provided in (22). Two different kinds of relative clauses (SRC and ORC) were examined. In (22), the relative clause that played soccer can be attached to either NP1 son or NP2 doctor. Depending on which NP the relative clause attaches to, it is called either High Attachment (HA) as in (22a) or Low Attachment (LA) as in (22b).

(21) a. the son1 of the doctor2[that played soccer] SRC
    b. the son1 of the doctor2[that I saw] ORC

(22) a. High attachment of (21a)
b. Low attachment of (21b)

The Korean counter part of (21) is in (23).

(23) a. [chwukkwu-lul han] uysa$_2$-uy atul$_1$ SRC
    soccer-Acc played doctor-of son
    ‘the son of the doctor that played soccer’

b. [nay-ka pon] uysa$_2$-uy atul$_1$ ORC
    I-Nom saw doctor-of son
    ‘the son of the doctor that I saw’

In Korean complex NP constructions, the -uy in (23) has been recognized as either a determiner (Lim 1981, Mok 2003), a case marker (Kim 2003,

(24) yepaywu-uy kacengpwu
    Actress-(?) maid

‘the maid of the actress’

Between them, An (2012, 2014)’s proposal that \(-uy\) is a linker between a modifier and its head noun was adopted in this paper. Under this proposal (both a case marker and a modifier), the syntactic representation of Korean complex NP constructions can be illustrated as in (25).

(25)

```
  DP
    NP
      FP maid
         DP -uy
             actress
```

Depending on the approaches to the usage of \(-uy\), the functional FP can be realized as a case phrase or a modifier phrase. In this paper, the details about the functional category of \(-uy\) will be put aside.
The annotated trees of (23) are presented in (26).

(26) a. High attachment of (23a)
b. Low attachment of (23a)
Once again, MGs syntax assumes that all languages are head initial languages and the word order of head final languages such as Korean is derived through movement\(^3\). The trees in this paper follow this assumption. Thus, the additional movements for the right word order take place.

The experimental studies investigating the attachment preference of the relative clause in NP complex constructions such as (21) and (23) have reported that there is cross-linguistic variation. LA is preferred in English (Phillips and Gibson 1997) but HA is preferred in Korean (Jun 2003, Lee and Kweon 2004, Ha 2005). With reference to these psycholinguistic studies, the metrics will be evaluated in the next section.

4.2 Results

The tables in (27) and (28) include all metrics that correctly predict low attachment preference in English and high attachment preference in Korean. The result of promotion analysis is in (27).

(27) The results (Promotion Analysis)

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Korean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SRC (21a)</td>
<td>ORC(21b)</td>
</tr>
<tr>
<td>HA vs LA</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HA vs LA</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgT</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgT(_I)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgT(_IU)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgT(_U)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT(_I)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT(_IU)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT(_U)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\(^3\) In MG parser system, the minimally paired structures are compared in a single language. The sentences were never evaluated cross-linguistically. The assumption that all languages are head initial languages is only for the technical convenience. Hence, this system does not imply that the movements that result in the right word order do not raise any additional processing difficulties in SOV languages.
Each metric of the basic metrics $\text{AvgT}$ and $\text{SumT}$ and the refined metrics $\text{AvgTI}$, $\text{AvgTIU}$, $\text{AvgTU}$, $\text{SumTI}$, $\text{SumTIU}$, and $\text{SumTU}$ predict the preference of relative attachment to a complex NP in all four different constructions correctly.

In English, only three nodes (who, $T$, $v'$) are non-trivial in both LA and HA constructions. The high tenure of these three nodes in HA is attributed to the structural complexity of the head noun. When the relative clause is attached to the high NP, the head NP of the relative clause is son of the doctor but when it is attached to the low NP, the head NP of the relative clause is just a single noun doctor. The complexity of head NP in HA construction makes it take longer to parse, so the three nodes (who, $T$, $v'$) stay longer in the memory stack and this causes the increase of the value of the tenure based metrics.

In Korean, when a relative clause is attached to a low NP, the relative clause is located in a relatively deep place, so the relative clause should go through more movement steps before the head noun is scanned. In addition, FP including the head noun doctor should stay long because FP is introduced in memory while the parser is seeking the relative clause. These raise the value of the tenure metrics. In other words, both LA preference in English and HA preference in Korean are related to the structural complexity.

Now let us take a look at the result of wh-movement analysis in (28).

(28) The results (wh-movement analysis)
As in (28), the basic metrics $\text{AvgT}$, $\text{MaxT}$ and $\text{SumT}$ and the refined metrics $\text{AvgT}_U$, $\text{MaxT}_U$, $\text{MaxT}^R$ and $\text{SumT}_U$ can predict the relative attachment preference in all constructions accurately. For the same reason in the promotion analysis, the LA preference is predicted in English. However, in Korean, the main reason why HA is preferred is different from the one in the promotion analysis. In the wh-movement analysis, only the highest node of the complex NP in the HA construction is conjectured by the parser before the relative clause is processed. However, in the LA construction, all lexical nodes in the complex NP are introduced into the memory stack before processing the relative clause. That is, the number of nodes which stay long in memory is bigger in LA constructions than in HA constructions. Due to that, the values of tenure based metrics are largely increased.

In sum, the basic tenure based metrics themselves correctly predict the relative clause attachment preference in each language. Like the results in dative argument attachment ambiguity, these results confirm that processing difficulty is caused by the structural complexity, not by the locality between the related elements.

5. The evaluation of the metrics including more data

This section will examine whether the MG parser can predict the preference reported in the wide range of different types of constructions simultaneously. The SRC preference between SRC vs ORC in Chinese, English, Japanese and Korean will be evaluated with the attachment ambiguities which we have seen in the previous sections. In this paper, the data provided in Graf et al. (2017) are used. The English example is in (29).

(29) a. The mayor [who invited the tycoon] likes wine. \hspace{1cm} SRC
b. The mayor [who the tycoon invited] likes wine. \hspace{1cm} ORC

Both promotion analysis and wh-movement analysis are evaluated respectively as well.

Before we consider the work in previous literature, let us see whether the
MG parser can correctly predict the processing effect on two different types of attachment ambiguity (dative argument attachment ambiguity and relative clause attachment ambiguity) in a comprehensive way.

(30) The results (promotion analysis)

<table>
<thead>
<tr>
<th>Dative attachment ambiguity</th>
<th>Relative clause attachment ambiguity</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: ambiguous matrix V</td>
<td>E: English K: Korean</td>
</tr>
<tr>
<td>U: unambiguous matrix V</td>
<td>S: SRC O: ORC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A_ID</th>
<th>A_DI</th>
<th>U_ID</th>
<th>U_DI</th>
<th>E_S</th>
<th>E_O</th>
<th>K_S</th>
<th>K_O</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvgT</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgTU</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumTU</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

(31) The results (wh-movement analysis)

<table>
<thead>
<tr>
<th>Dative attachment ambiguity</th>
<th>Relative clause attachment ambiguity</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: ambiguous matrix V</td>
<td>E: English K: Korean</td>
</tr>
<tr>
<td>U: unambiguous matrix V</td>
<td>S: SRC O: ORC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A_ID</th>
<th>A_DI</th>
<th>U_ID</th>
<th>U_DI</th>
<th>E_S</th>
<th>E_O</th>
<th>K_S</th>
<th>K_O</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvgT</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgTU</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumTU</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In both analyses, the same metrics (AvgT, AvgTU, SumT, SumTU) predict the preference in two different attachment ambiguity constructions. They are all tenure based metrics. This suggests two things. First, the non-trivial nodes’ tenure is mainly related to processing of the attachment ambiguity in general. Second, the attachment ambiguities could be explained uniformly with the very restricted number of the tenure based metrics regardless of the analyses.
Taking a look at the metrics in (30) and (31) again, the metrics giving the right prediction consist of two basic metrics (M) and their refined metrics (MU). In other words, the metrics which are applied to only unpronounced leaf nodes play a decisive role in predicting the preference as well. It is conjectured that the unpronounced heads in the relative clauses such as C and Rel become heavier than the ones in the preferred structures, due to movements such as scrambling in a relative clause of dative argument ambiguity. As it is mentioned in the previous sections, since the number of non-trivial nodes in the attachment ambiguity is the same, the result of AvgT and AvgTU are obvious.

The results in (30) and (31) show that the very limited number of metrics capture the processing effect in attachment ambiguity. In order to examine whether these restricted metrics can capture the processing effect in the various constructions, each attachment ambiguity and RC processing effect in previous work will be evaluated at first and then the all constructions will be evaluated together.

The results of metrics evaluation for dative attachment ambiguity and processing RCs for promotion analysis are in (32). The relevant metrics only are provided in (32). As you can see, none of metrics makes the right prediction. The metrics predicting the preference of dative argument ambiguity fail to predict the preference of relative clause processing. In particular, they predicted the opposite preference in Korean relative clause processing.

(32) The results (promotion analysis)

<table>
<thead>
<tr>
<th>Dative attachment ambiguity</th>
<th>Relative clause processing SRC vs ORC</th>
</tr>
</thead>
<tbody>
<tr>
<td>U: unambiguous matrix V</td>
<td></td>
</tr>
<tr>
<td>A_ID A_DI U_ID U_DI</td>
<td>E C K J</td>
</tr>
<tr>
<td>AvgT</td>
<td>Yes Yes Yes Yes Yes Yes Yes No No Yes</td>
</tr>
<tr>
<td>AvgTU</td>
<td>Yes Yes Yes Yes Yes Yes Yes No No Yes</td>
</tr>
<tr>
<td>BoxTI</td>
<td>Yes Yes Yes Yes Yes Yes Yes No Yes</td>
</tr>
</tbody>
</table>
The same thing is observed in the results of wh-movement analysis in (33). Except MaxS\(^R\), all metrics which successfully predict the preference in dative attachment ambiguity fail to predict East Asian languages’ RC processing preference.

(33) The results (wh-movement analysis)
Then, why does the metric MaxSR which succeeds to predict the preference in the wh-movement analysis fail to predict the preference in the promotion analysis? Indeed, for the promotion analysis, MaxSR failed to predict the preference only in the construction U-DI (unambiguous matrix verb and DO-IO base word order) as in (32). MaxSR is the recursive application of MaxS which measures the movement distance of a node. In the other constructions, either the second ranked value or the third ranked value is bigger in the relative clause verb attachment construction than in the matrix verb attachment construction. However, in the construction U-DI, the value of the node which goes through the longest movement is bigger in the matrix verb attachment construction than in the relative clause verb attachment construction.

Now, let us take a look at the result of the relative clause attachment ambiguity and relative clause processing effect. As shown in (34) and (35), none of metrics predicts the preference correctly. The metrics predicting the preference of the relative clause attachment ambiguity fail to predict the preference of relative clause processing in the results of both analyses.

(34) The results (promotion analysis)

<table>
<thead>
<tr>
<th></th>
<th>Relative clause attachment ambiguity</th>
<th>Relative clause processing SRC vs ORC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A_ID A_DI U_ID U_DI</td>
<td>E: English C: Chinese K: Korean J: Japanese</td>
</tr>
<tr>
<td>AvgT</td>
<td>Yes       Yes     Yes     Yes</td>
<td>E: Yes C: No K: No J: No</td>
</tr>
<tr>
<td>AvgT₁</td>
<td>Yes       Yes     Yes     Yes</td>
<td>E: Yes C: No K: Yes J: No</td>
</tr>
<tr>
<td>AvgT₂</td>
<td>Yes       Yes     Yes     Yes</td>
<td>E: Yes C: No K: No J: No</td>
</tr>
<tr>
<td>AvgT₃</td>
<td>Yes       Yes     Yes     Yes</td>
<td>E: Yes C: No K: No J: No</td>
</tr>
<tr>
<td>BoxTₚ</td>
<td>Tie       Tie     Tie     Tie</td>
<td>E: Yes C: Yes K: Yes J: Yes</td>
</tr>
<tr>
<td>MaxSₘ</td>
<td>Tie       Tie     Tie     Tie</td>
<td>E: Yes C: Yes K: Yes J: Yes</td>
</tr>
<tr>
<td>MaxTₘᵢP</td>
<td>Tie       Tie     Tie     Tie</td>
<td>E: Yes C: Yes K: Yes J: Yes</td>
</tr>
<tr>
<td>MaxTₘᵢₚ</td>
<td>Yes       Tie     Yes     Yes</td>
<td>E: Yes C: Yes K: Tie J: Yes</td>
</tr>
<tr>
<td>MaxTₘᵯₚ</td>
<td>Yes       Tie     Yes     Yes</td>
<td>E: Yes C: Yes K: Tie J: Yes</td>
</tr>
<tr>
<td>MaxTₘᵯₚᵤ</td>
<td>Yes       Tie     Yes     Yes</td>
<td>E: Yes C: Yes K: Yes J: Yes</td>
</tr>
<tr>
<td>SumS</td>
<td>Tie       Tie     Yes     Yes</td>
<td>E: Yes C: Yes K: Yes J: Yes</td>
</tr>
</tbody>
</table>
As seen in the tables in (32, 33, 34, and 35), the four metrics which succeed in predicting the preference in both attachment ambiguities failed to predict the relative clause processing effect. Then, the following question is raised: Is it still possible to predict human sentence processing preference of three different types of constructions (dative attachment ambiguity, relative clause attachment ambiguity, and relative clause processing effect) uniformly?
Since it is impossible for a single metric to correctly predict the processing effect on the three different constructions, the ranked metrics in an OT-like manner in Graf et al. (2017) is utilized. The following tables in (36) and (38) include the results from all constructions. Only the relevant metrics are presented there.

(36) The results (promotion analysis)

<table>
<thead>
<tr>
<th>Dative attachment ambiguity</th>
<th>Relative clause attachment ambiguity</th>
<th>Relative clause processing SRC vs ORC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: ambiguous matrix V</td>
<td>E: English</td>
<td>E: English</td>
</tr>
<tr>
<td>U: unambiguous matrix V</td>
<td>K: Korean</td>
<td>C: Chinese</td>
</tr>
<tr>
<td>S: SRC</td>
<td>O: ORC</td>
<td>K: Korean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J: Japanese</td>
</tr>
<tr>
<td>A_ID</td>
<td>A_DI</td>
<td>U_ID</td>
</tr>
<tr>
<td>AvgT</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AvgTU</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MaxTR_{IP}</td>
<td>Tie</td>
<td>Tie</td>
</tr>
<tr>
<td>SumT</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT_{IP}</td>
<td>Tie</td>
<td>Tie</td>
</tr>
<tr>
<td>SumTU</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In promotion analysis, the basic metrics SumT and AvgT which successfully predicted the preference in dative attachment ambiguity and relative clause attachment ambiguity, failed to predict the processing preference of relative clauses in East Asian languages. SumT simply failed to predict a preference in Korean relative clause processing. It actually predicts the opposite result (ORC preference) of the psycholinguistic result (SRC preference). This shows that tenure is not enough to uniformly account for human processing preference even in one language.

When ranked metrics are taken into consideration, the combinations of the metrics AvgT, AvgTU, MaxTR_{IP}, SumT, SumT_{IP} and SumTU can predict the preference in all constructions as in (37).
(37) Ranked metrics

\[
\langle \text{SumT}_{IP}, \text{SumT} \rangle, \langle \text{MaxT}^R_{IP}, \text{SumT} \rangle \\
\langle \text{SumT}_{IP}, \text{SumT}_U \rangle, \langle \text{MaxT}^R_{IP}, \text{SumT}_U \rangle \\
\langle \text{SumT}_{IP}, \text{AvgT} \rangle, \langle \text{MaxT}^R_{IP}, \text{AvgT} \rangle \\
\langle \text{SumT}_{IP}, \text{AvgT}_U \rangle, \langle \text{MaxT}^R_{IP}, \text{AvgT}_U \rangle
\]

The number of combined metrics which can predict the psycholinguistic result correctly is also very restricted (8 out of 1600) and all related metrics are based on tenure. This shows that the processing effect on several different constructions can be explained by their structural complexity with very limited combinations of metrics. Interestingly, Korean SRC preference, which required to consider hierarchical distance between fillers and gaps, is also predicted accurately with the combination of selected tenure based metrics.

At last, let us consider wh-movement analysis.

(38) The results (wh-movement analysis)

<table>
<thead>
<tr>
<th></th>
<th>Dative attachment ambiguity</th>
<th>Relative clause attachment ambiguity</th>
<th>Relative clause processing SRC vs ORC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A: ambiguous matrix V</td>
<td>E: English</td>
<td>E: English</td>
</tr>
<tr>
<td></td>
<td>U: unambiguous matrix V</td>
<td>K: Korean</td>
<td>C: Chinese</td>
</tr>
<tr>
<td></td>
<td>A_ID</td>
<td>A_DI</td>
<td>U_ID</td>
</tr>
<tr>
<td>AvgS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MaxS^R</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SumT</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SumT_U</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

(39) Ranked metrics

\[
\langle \text{SumT}, \text{AvgS} \rangle, \langle \text{SumT}_U, \text{AvgS} \rangle \\
\langle \text{SumT}, \text{SumS} \rangle, \langle \text{SumT}_U, \text{SumS} \rangle \\
\langle \text{SumT}, \text{MaxS}^R \rangle, \langle \text{SumT}_U, \text{MaxS}^R \rangle
\]
As in (39), wh-movement analysis result shows that size metrics AvgS, SumS, and MaxS\textsuperscript{R} as well as tenure metrics SumT and SumT\textsubscript{U} should be considered together in order to predict both the preferences of attachment ambiguity and relative clause processing effects simultaneously. From a linguistic view, the combination of tenure metrics and size metrics has been argued to reflect the intuition that the memory load with the length of movement dependencies is closely related to the processing difficulty (Graf et al. 2015, 2017). The evaluation result in (39) confirms this claim for a much larger dataset.

In sum, the types of metrics used for the correct prediction of relative clause processing preference are different depending on the analysis, such as tenure metrics for promotion analysis but size metrics for wh-movement analysis. Each combination of the selected metrics tells us what kind of structural complexity are related to processing sentences. In addition, the order of metrics is different. In the promotion analysis, the metrics for relative clauses should be applied earlier than the other metrics but those should be applied later than the others in wh-movement analysis. The relation between the order of combined metrics and processing effect still remains to be explained.

6. Conclusion

This paper examined how an MG model predicts the processing preference reported in the psycholinguistic literature by using the memory usage metrics developed by Graf et al. (2017) as they are related to several different constructions. The results suggest that the processing difficulty can be explained as a consequence of structural complexity, thus explaining processing preferences that the Active Filler Strategy (Frazier 1987) and Recency (Gibson et al. 1998), both locality-based theories, are unequipped to handle. In general, under the computational linguistic approach using MG parsing model, if the structure is more complex, it causes more memory usage.

The data tested in this paper show that hierarchical complexity should necessarily be considered. However, more counter-examples exist showing
the linearity effect on sentence processing regardless of the hierarchical locality. One of them is the embedded wh-scope preference. In the exact same construction in the dative attachment ambiguity, when the dative is a wh-word, the processing effect is opposite to when the dative is a regular NP (Aoshima et al. 2004). This processing contrast in the same construction depending on the different types of lexical items tells us that the property of lexical items such as scope bearing elements such as wh-words, *some*, *every* and so on can cause the different level of syntactic complexity. How an MG parser can handle the processing effect apparently caused by locality based effects is a topic for future research.

References


