[Proceedings of the Society for Computation in Linguistics](https://scholarworks.umass.edu/scil)

Volume 6 Article 23

6-1-2023

Processing Advantages of End-weight

Lei Liu
Universitat Leipzig, lei.liu@uni-leipzig.de

Recommended Citation

Liu, Lei (2023) "Processing Advantages of End-weight," *Proceedings of the Society for Computation in Linguistics*: Vol. 6, Article 23.
DOI: https://doi.org/10.7275/7etw-rj24
Available at: https://scholarworks.umass.edu/scil/vol6/iss1/23

Processing Advantages of End-weight

Lei Liu
Institute of Linguistics
Leipzig University
lei.liu@uni-leipzig.de

Abstract

Previous research has established that English end-weight configurations, where sentence components of greater grammatical complexity appear at the ends of sentences, demonstrate processing advantages over alternative word orders. To evaluate these processing advantages, I analyze how a Minimalist Grammar (MG) parser generates syntactic structures for different word orders. The parser's behavior suggests that end-weight configurations require fewer memory resources for parsing than alternative structures. This memory load difference accounts for the end-weight advantage in processing. The results highlight the validity of the MG processing approach as a linking theory connecting syntactic structures to behavioral observations. Additionally, the results have implications on the structure and processing of languages where an "initial-weight" is preferred.

1 Introduction

The grammatical weight of a phrase has consequences on sentence processing. One observable consequence is word order preference. In English, a direct object (DO) typically follows the verb immediately. When the DO is heavy, the language allows an otherwise awkward order, where the heavy DO occurs at the end.

(1) a. Emma explained [$_{\text{DO}}$ the regulations] to [$_{\text{IO}}$ Jim].
  b. Emma explained to [$_{\text{IO}}$ Jim] [$_{\text{DO}}$ all the regulations regarding import and export taxes for pottery].
  c. ? Emma explained to [$_{\text{IO}}$ Jim] [$_{\text{DO}}$ the regulations].
  (Stallings and MacDonald, 2011)

Sentence (1a) shows the order Verb-DO-Indirect Object (IO). This order is considered natural when compared with Verb-IO-DO in (1c). But when the DO is complex – e.g., containing a complex modifier – a Verb-IO-DO order (1b) becomes possible, if not preferred. Sentences such as (1b) are known as heavy NP shift (HNPS) sentences.

A similar end-weight preference is found in English particle verb (PV) constructions. In a PV construction, the particle can either occur right next to the verb (the joined order) or be separated from the verb by the object (the separated order). When the object is heavy, the joined order is preferred. This is illustrated in (2).

(2) a. ... I **looked up** [a person who answered a query I posted on the internet]...
  b. *I **looked** [a person who answered a query I posted on the internet] **up**...
  (Cappelle, 2005, 19)

Despite clear intuitions of end-weight preferences in the above examples, the definition and measurement of grammatical weight are controversial. Without getting too much into each proposal[1], two things stand out as important for understanding grammatical weight, a) the structural information of the heavy phrase is a better measurement of weight than counts on linear strings (e.g., number of words, phrases) (among others, Ross, 1986; Hawkins, 1994; Wasow, 1997), and b) compared with the weight of a single phrase, the relative weight of sentence components better predicts processing phenomena (Hawkins, 1994; Wasow, 1997; Stallings and MacDonald, 2011).

In this study, I explore whether these weight-related processing phenomena follow from the corresponding syntactic structures. Specifically, a top-down parser for Minimalist Grammars (MG) is used to build HNPS and PV constructions and their word order alternatives. Based on how the parser

[1]The readers are refereed to Chapter 2 of Liu (2022) for a brief review of weight measurements and Chapter 2 of Wasow (2002) for a discussion of some of the proposals under experimental/corpus settings.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 250-258.
Amherst, Massachusetts, June 15-17, 2023

traverses each syntactic tree, a set of complexity metrics measures memory resource allocation in the tree-building processes, from which we can infer the processing difficulties of each word order.

To apply this MG parsing approach, it is necessary to define the MG implementations of the relevant syntactic proposals and to establish the complexity metrics based on the parser's behavior. These are discussed in Section 2. Section 3 presents modeling results. Section 4 discusses the implications of the current results on the apparent opposite preference for weight configuration, "initial weight", observed in languages like Japanese.

To preview the results, the parsing model suggests that the preferences for HNPS and joined PV constructions follow from the processing difficulties associated with the syntactic structure of competing word orders. The results strengthen the validity of the MG parsing approach as a linking theory connecting structural proposals to behavioral data. The results also broaden the empirical coverage of the processing phenomena the MG parsing approach is shown to successfully capture (e.g., center- vs. right-embedding (Kobele et al., 2013); subject vs. object relative clause in various languages (Graf et al., 2017), attachment ambiguity in English and Korean (Lee, 2018), gradient of difficulty in Italian relative clauses (De Santo, 2019)).

2 MG Parsing

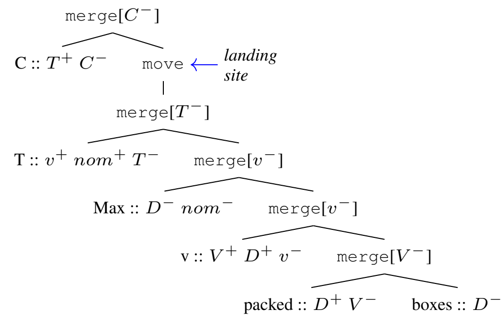
On an intuitive level, the MG parsing model used in this study infers processing difficulties of a given sentence according to how memory-costly it is to parse by a parser for MG.

2.1 Minimalist Grammar

Minimalist Grammar (Stabler, 1996) is a lexicalized, context-sensitive formalism incorporating the Minimalist Program (Chomsky, 2014). Such incorporations allow the MGs to relatively straightforwardly represent Minimalist syntactic proposals. In MGs a grammar is a set of lexical items, which are expressed in feature bundles containing information including pronunciation, category, movement dependencies, etc. Similar to the standard Minimalist Program-styled derivation, these lexical items are built into sentences (trees) via `merge`, which combines lexical items and/or phrases; and `move`, which regulates movements.

To illustrate, (3) is a toy MG derivation tree for the sentence *Max packed boxes*.

(3)

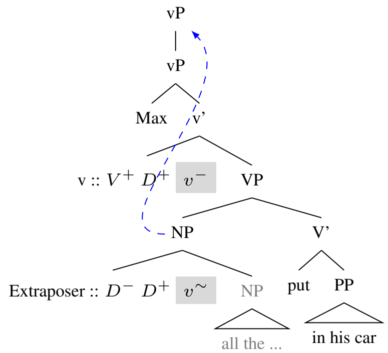


In (3), the uppercase features $X^{\pm}$ are `merge` features. The superscripts $^+$ and $^-$ indicate selector and category for merge, respectively. For instance, in the bottom of the tree, *packed ::* D^+ V^- merges with *boxes ::* D^- and "checks" the matching D feature. The lowercase features $y^{\pm}$ are `move` features with the superscripts $^+$ and $^-$ representing licensor and licensee. Again in (3), the subject movement is indicated with matching nominative features nom^+ and nom^-. The movement also creates a unary branching at the landing site, while the mover remains in its merge position. This creates an order mismatch between the leaf nodes and the linear string, which will become important when we discuss MG parsing.

In addition to standard `merge` and `move` operations, MGs comfortably allow rightward movement, an operation proposed for deriving HNPS, among other things. Torr and Stabler (2016) show that MGs can be extended to allow rightward movement without affecting the weak expressive power. The authors derive rightward movement with a null extraposer bearing rightward movement licensee feature $x^{\sim}$. The extraposer merges with the shifting constituent and shifts with it rightward to category x. For instance, an extraposer causing the heavy NP to move rightward and adjoin to the vP has the feature bundle in (4).

(4) Extraposer :: D^- D^+ $v^{\sim}$

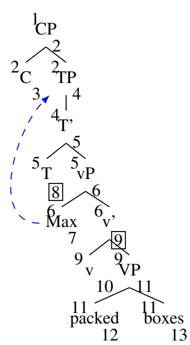
The null extraposer selects the heavy NP, further projects an NP, and shifts to the right of the nearest vP category. (5) schematizes the derivation tree for rightward movement. The matching rightward movement feature pair is highlighted in shade.

(5) 

Despite its now unpopularity due to alternative proposals such as Kayne's 1994 Linear Correspondence Axiom (LCA), rightward movement is not discarded by the MGs on formal grounds. More broadly, this shows how closely MGs can incorporate "devices from the syntacticians' toolbox" (Graf et al., 2017), which for our purpose also sets the stage for parsing.

2.2 MG parser and complexity metrics

A top-down MG parser (Stabler, 2013), intuitively, takes as input a sentence represented in a string of words and builds the structure based on a set of MG rules following a top-down, left-to-right order. For example, (6) is an annotated derivation tree showing a correct parse for the sentence *Max packed boxes*. Merge and move nodes are replaced with their names for better readability.

(6) 

Following conventions in MG parsing studies (Kobele et al., 2013; Graf et al., 2017), the numbers on the two corners of each node in (6) indicate steps at which the node is conjectured (superscripted numbers, or *indices*) and confirmed (subscripted numbers, or *outdices*) by the parser.

Crucially, the top-down MG parser is defined to be able to temporarily not follow the left-to-right tree traversal in order to find the leftmost word in the current linear string. This allows the parser to handle the leaf nodes and linear string mismatch. For instance, in (6), the subject movement alters the linear order of T head and the subject, *Max*. Assuming the parser has correctly conjectured a TP (step 4), and that the mover comes from the right branch (step 5), it is defined to then go right (step 6) to find the mover while putting in memory every node it conjectures along the way (in this case, only the T node). After the parser finds and confirms the mover, it goes back to work on the nodes that were stored in memory. The difference of index and outdex on T thus reflexes how many steps the node was stored in memory. This contrasts with a typical top-down parser for CFG, in which case a parse is abandoned when there's an order mismatch between the leaf nodes and the linear string. Moreover, for a successful parse where the leaf nodes and the linear string align, the number of steps a node is held in memory depends solely on the size of its left sister, which is not as informative.

Using the indices and outdices on a derivation tree, it is possible to infer the memory usage of the parser as it builds the tree, based on which we can model sentence processing difficulties. The MG parsing model distinguishes several measurements of memory usage based on the indices and outdices (Graf et al., 2017). Among the measurements, tenure measures how long a node is kept in memory. The idea of measuring processing load based on the "time" an item is kept in memory is discussed in psycholinguistics literature such as Gibson (1998) among others. And this idea is formalized in Joshi (1990); Rambow and Joshi (2015) and the line of work on modeling human sentence processing using Tree Adjoining Grammar (TAG). More recent work has shown that tenure-related metrics make reliable processing predictions cross-linguistically (among others, Kobele et al., 2013; Graf et al., 2017; De Santo, 2019). Based on tenure, a large set of complexity metrics can be defined. Here we focus on MaxT, defined in Kobele et al. (2013) as the following: $\texttt{MaxT} = max(\text{tenure-of}(n))$. In other words, MaxT measures the maximum number of steps any node is kept in memory.

With complexity metrics such as MaxT, we can already address one of the two key points to understand grammatical weight – structure information is a better measurement for weight than word count. Structural characterization of weight determines grammatical weight based on the syntax rather than the number of words or phrases. Consider the following two sentences.

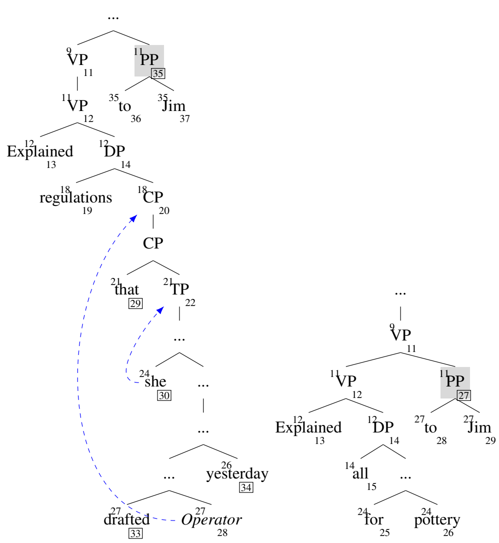
(7) a. Emma explained [all the regulations

that she drafted yesterday] to [Jim].

(adapted from Stallings and MacDonald (2011))

b. Emma explained [all the regulations regarding taxes for pottery] to [Jim].

The objects in (7) are both seven words long. But the one in (7a) contains a relative clause (RC), which adds extra processing difficulties (Fraser, 1966; Ross, 1967). Assuming a *wh*-movement analysis (Chomsky, 1977) for RCs, Figure 1 are excerpts of annotated derivation trees showing how the MG parser builds structures for the sentence pair.



(a) relative clause tree (b) long NP tree

Figure 1: Excerpts of derivation trees for the sentence pair in (7).

The RC-modified DP in the tree on the left takes the parser 22 steps to build, compared to 14 steps for the DP in the right tree which has the same length but no RC modification. This results in an overall `MaxT` difference of 24 vs. 16 (on the shaded nodes), predicting that (7a) is more difficult to parse because of its more complex syntax due to RC modification. This tenure-based prediction is also how we model processing differences between end-weight structures and their word order alternatives, which we discuss below.

3 Modeling Results

To evaluate processing advantages of end-weight structures, I compare these structures in a pairwise fashion with their word order alternatives. Consistent with previous work, each comparison is between two correctly constructed trees. That is, the parer is assumed to be deterministic and always finds the right parse. Any potential processing load associated with ambiguity and reanalysis is factored out. This methodological choice highlights the role of syntactic structure in predicting processing loads in different weight configurations, which is exactly what we set out to explore.

3.1 End-weight in HNPS

For HNPS, the comparisons are between the object shift order and the canonical order. A total of four pairs of sentences are used in the comparisons:

(8) a. Max put [$_{\text{DP}}$ boxes] [$_{\text{PP}}$ in a car]. (short-DP short-PP)

b. Max put [$_{\text{PP}}$ in a car] [$_{\text{DP}}$ boxes]. (short-PP short-DP)

(9) a. Max put [$_{\text{DP}}$ boxes] [$_{\text{PP}}$ in a car made in Stuttgart]. (short-DP long-PP)

b. Max put [$_{\text{PP}}$ in a car made in Stuttgart] [$_{\text{DP}}$ boxes]. (long-PP short-DP)

(10) a. Max put [$_{\text{DP}}$ all the boxes of home furnishings] [$_{\text{PP}}$ in a car]. (heavy NP)

b. Max put [$_{\text{PP}}$ in a car] [$_{\text{DP}}$ all the boxes of home furnishings]. (heavy NP shift)

(11) a. Max put [$_{\text{DP}}$ all the boxes of home furnishings] [$_{\text{PP}}$ in a car made in Stuttgart]. (long-DP long-PP)

b. Max put [$_{\text{PP}}$ in a car made in Stuttgart] [$_{\text{DP}}$ all the boxes of home furnishings]. (long-PP long-DP)

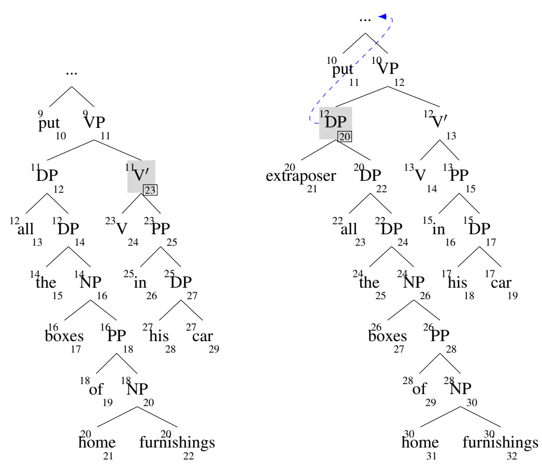
Given the behavior data, we expect the parser to predict an object shift advantage *only* for the pair in (10) ((10b) is advantageous). The pair in (8) contains no heavy constructions, thus no shift advantage is expected. The pair in (9) contains heavy PPs, but the canonical order is the end-weight order, no shift advantage is expected. Moreover, if there is a relative weight effect, i.e., the shift order is only preferred when the object is much more complex than other sentence components, we expect to see no shift order advantage for the pair in (11), where both DPs and PPs are complex.

Table 1 summarizes the parser's prediction for each weight condition. Overall, `MaxT` predicts expected processing preferences in all weight configurations: the shift order has a processing advantage only when the object DP is complex – in fact, more

Weight config.	Shift advantage?	Parser prediction
Both light	No	No
Heavy PP	No	No
Heavy NP	Yes	Yes (`MaxT`: 8 vs. 12)
Both Heavy	No	No (`MaxT`: 14 vs. 12)

Table 1: Summary of the predictions for each weight configuration in object shift constructions

complex than the PP. This is clearer if we look at the annotated derivation trees in Figure 2.



(a) canonical order (b) HNPS order

Figure 2: Excerpts of derivation trees for canonical word order (2a) and HNPS order derived via rightward movement (2b)

First, `MaxT` found on the shaded nodes predicts a HNPS advantage. If the heavy NP does not move, the parser would have to fully build the heavy NP until it can go back to the earlier branch to continue work on V'. This causes a great tenure on the V' node as shown in Figure 2a. In contrast, the rightward movement alters the linear order of the two branches and essentially makes the parser to delay the heavy lifting of building the NP and work first on the right branch. Since the size of the right branch is much smaller than its sister, first working on this branch means less waiting time for the left branch compared to the opposite order. This results in a smaller `MaxT`, as shown in Figure 2b. A `MaxT` difference of 8 vs. 12 predicts a HNPS advantage.

It is also transparent to anticipate the relative weight effect from Figure 2. As the right sibling of the heavy NP, or in fact, the lower PP grows in complexity, the shifted order would no longer be preferred based on `MaxT`. Indeed, under the condition where both DP and PP are complex (i.e., (11)), the shifted order has a higher `MaxT` (14 vs. 12), predicting that it is no longer advantageous.

3.2 End-weight in PV

English particle verb construction can be thought of as an extreme case of relative weight, because the object is always comparing with a one-word particle. If the prediction about relative weight is true, that it is advantageous for processing to put the relatively complex sentence components at the sentence end, a joined order for a PV should always be preferred over a separated order. To give away the results, the MG parser indeed prefers a joined order irrespective of DP length. This has interesting implications on how to interpret MG models. We will pick this up after presenting the modeling results.

Similar to the processing model for HNPS, a total of three pairwise comparisons were made between joined order and separated order for a PV construction. For each word order, three DP conditions were included: short DP (2 words), long DP with prenominal modifiers ([mod-DP], 7 words), and long DP with post-nominal modifiers ([DP-mod], 7 words):

(12) short DP
   a. Chris **put on** a hat.
   b. Chirs **put** a hat **on**.

(13) [mod-DP]
   a. Chris **put on** a very very very very expensive hat.
   b. Chirs **put** a very very very very expensive hat **on**.

(14) [DP-mod]
   a. Chris **put on** a hat which Alex made with love.
   b. Chris **put** a hat which Alex made with love. **on**.

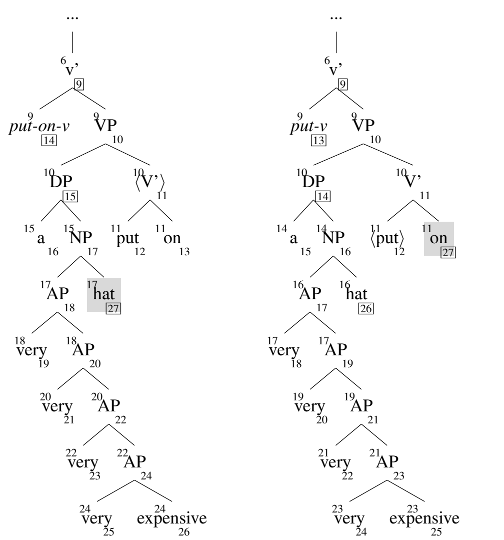
The contrast between short and long DPs helps demonstrate a potential end-weight advantage. The contrast in two long DP conditions is to confirm the role structure plays in measuring grammatical weight. It also tests the claim that for a subset of PVs, the location of DO modifiers makes a processing difference (Lohse et al., 2004). For space and cohesiveness reasons, we will not discuss the results of this PV subset.

Assuming a particle stranding analysis for separated PVs, and a complex verb raising one for the joined order (Larson, 1998; Johnson, 1991), Table 2 summarizes the parser's prediction for each DP condition of the PV constructions. Overall, MaxT predicts that a joined order is easier to parse than a separated one under all weight configurations.

Weight config.	**Joined advt?**	**MG parser**
Short DP	No/Unclear	Yes (MaxT 5 vs. 6)
[mod-DP]	Yes	Yes (MaxT 10 vs. 16)
[DP-mod]	Yes	Yes (MaxT 8 vs. 24)

Table 2: Summary of the predictions for each weight configuration in particle verb constructions

We first take a look at the end-weight configuration. For instance, the parser builds the two PV orders under the [mod-DP] condition as shown in Figure 3.

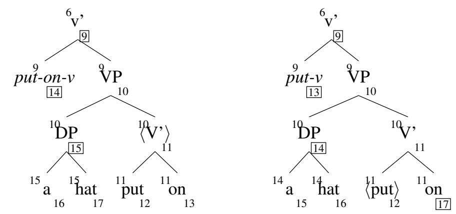


(a) **put on** a very very...hat. (b) **put** a very very...hat **on**.

Figure 3: Excerpts of derivation trees for PV joined order derived via complex verb raising (3a) and PV separated order derived via particle stranding (3b)

In the structure building process, the parser conjectures the particle at the same step when it conjectures the verb (step 11). For a joined order (left), the particle is confirmed and flushed out of the memory after the verb (step 13). For a separated order (right), the particle is held in memory until the long DP is fully built. This is memory costly and is where MaxT is found.

Furthermore, for the separated order the particle is always held in memory for some time during the parse, irrespective of the DP size. This predicts that a separated order is almost always disfavored over a joined order based on tenure. Figure 4 shows a joined order and separated order derivation for short DPs. Under this condition, the extra tenure on the particle of the separate order still makes it more difficult to parse than the joined order. This is unintuitive given that their corresponding sentences in (12) sound equally natural. I will come back to this briefly in Section (4).

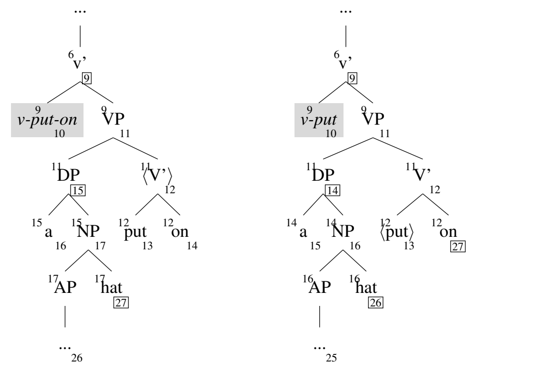


(a) **put on** a hat. (b) **put** a hat **on**.

Figure 4: Excerpts of derivation trees for PV orders under the short DP condition

The syntactic assumption for joined and separated PV orders both involve head movement, that of a complex verb and a verb head, respectively (indicated in the figures with angle brackets). In Figure 3 the landing site of head movement is assumed to be on the left of the v head, following Adger (2003). When discussing serial verbs in German and Dutch, Kobele et al. (2013) note that when an MG parser builds structures with head movements, the landing site of the head movements affects memory recourse allocation. Since v head is silent, head movement landing on the right of v is string equivalent to when landing on the left. So additional comparisons were made assuming the opposite landing site to see a potential processing effect. An excerpt of the derivation trees is in Figure 5.

The landing site of head movement does make a difference in memory cost, but the difference does not affect preference predictions. From the parser's perspective, if the landing site is to the right of the little *v* head, the parser can conjecture and confirm the empty *v* head right away (at step 10 in Figure 5). This contrasts with when the landing site is to the left (Figure 3 and 4), in which case the parser will have to confirm the verb head/complex verb before confirming the little *v* head. This causes tenures on the little *v* head for both orders (trivially different by one step due to the particle). For both directions of the landing site, the memory resources needed for building v are almost identical between separated and joined orders. Processing predictions are



(a) V-to-v on the right - joined order (b) V-to-v on the right - separated order

Figure 5: Excerpts of derivation trees for PV orders assuming V-to-v movement landing on the right

unaffected – under both head movement conditions, `MaxT` is constantly lower for the joined order.

4 Discussions

In this paper, I have shown that the processing advantages of end-weight structures such as HNPS and PV joined order follow from their corresponding syntactic structure: an end-weight structure is more memory efficient to parse. We arrive at this conclusion by utilizing MG processing models which link syntactic structures to behavioral observations based on a psycholinguistically well-motivated factor, memory. The results presented in this study widen the collection of the empirical phenomena the parsing model can capture. Furthermore, given the rigorous link that underlines the MG processing model, one can make syntactic predictions based on behavioral data. This is briefly illustrated below concerning the apparent opposite weight preference: the initial weight, or long-before-short preference observed in Japanese (Yamashita and Chang, 2001).

Japanese is an SOV, head-final language. When the object becomes long, it tends to appear at the beginning of a sentence, contrary to English HNPS (Yamashita and Chang, 2001).

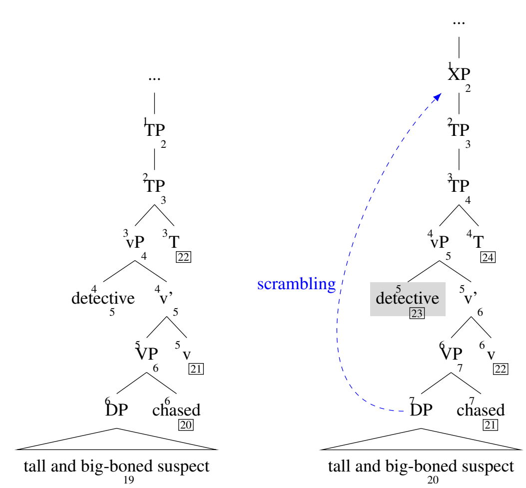
(15) a. [$_O$ Se-ga takakute gassiri sita
height-nom tall-and big-boned
hanni-o]$_i$ [$_S$ Keezi-ga] t_i Oikaketa.
suspect-acc detective-nom chased
'The detective chased the suspect who is tall and big-boned.'

b. [$_S$ Keezi-ga] [$_O$ hanni-o]
detective-nom suspect-acc
Oikaketa.
chased
'The detective chased the suspect.'

(adapted from Yamashita and Chang (2001))

Syntactically, object shift such as (15a) is often considered a case of scrambling. A great number of proposals have been made on scrambling cross-linguistically (Ross 1986; Saito 1992; Miyagawa 1997; Bošković and Takahashi 1998; Bailyn 2001, among others). The proposals can be roughly categorized into movement-based derivation and base-generation. The movement-based analyses (e.g., Saito, 1992; Miyagawa, 1997) argue that the scrambled constituent moves leftward and adjoins to a high specifier position. The base-generation analysis (Bošković and Takahashi, 1998; Bošković, 2004), on the other hand, base-generates the "scrambled" constituent which then checks relevant features in an obligatory LF lowering.

For our processing model, movement-based derivations do not derive an initial weight preference. Suppose the parser takes 13 steps to build the heavy object NP, which is roughly the steps needed to build the long object in (15a), depending on one's analysis of prenominal relative clauses. We compare excerpts of the derivation trees for canonical word order and shifted word order in Figure 6.



(a) Japanese SOV order with heavy object (b) Japanese OSV order derived via scrambling

Figure 6: Parsing heavy object structures in Japanese

Figure 6 shows that a shifted structure (6b) is more difficult to parse in the current parsing model. This is because the scrambled object linearly precedes but is structurally beneath the subject. This means that the parser first conjectures the subject, but needs to hold it in memory, find and build the object, before it can finally return to build the object. This comes with great memory cost, making the initial weight structure difficult to parse, contrary to behavior observations.

There are two possibilities to potentially reconcile the typological difference of where to put heavy constituents. First, the unexpected processing prediction for the OSV order in Japanese could be due to the syntactic assumption, i.e., the object shift analysis. For the object shift analysis, memory burdens arise when linear and structural orders do not match. If the DP merges high in the structure, as suggested by the base-generation analysis, The structural relation and linear order of the object and subject are aligned. The parser would then build the "scrambled" structure first without holding the subject in memory.

Second, it could be the case that an initial weight is preferred for non-syntactic reasons. The link the MG parsing model establishes is one between syntactic structure and behavioral data. If the current syntactic assumption is well-motivated but cannot make correct behavioral predictions, one is prompted to look for non-syntactic reasons. Indeed, Yamashita and Chang (2001) argue that languages order their constituents depending not only on the syntactic form but also on the salience. For Japanese, the salience of a heavy constituent combined with a word order that is less restrictive than in English results in an initial weight preference in Japanese.

It is beyond the scope of this paper to fully test out these possibilities. The claim made here is a methodological one. On the one hand, the MG parsing models show how syntactic analyses impact processing predictions in a quantitative, structure-based way. When the processing phenomena are clear, the parsing models are useful in evaluating syntactic proposals. Such applications have been reported in Liu (2018) where a rightward movement structure predicts a HNPS advantage while requiring the fewest assumptions on memory cost calculations among competing structures like remnant movement and PP movement; and in Pasternak and Graf (2021) who verifies and broadens the processing predictions of an unbounded, cyclic QR analysis for scope interpretation.

On the other hand, by taking seriously the syntax and its processing predictions, the MG models shed light on multi-factorial analyses of processing preference. In Section (3.2) we saw that a joined PV order is almost always favored by the parsing model, which might seem unintuitive. However, based on a speech production experiment, Dehé (2002) reports a preference for joined order and attributes the preference to the neutral, default status of the joined order. Our processing model might offer one way to understand this default status: the default structure is the one that is easy to process.

Similarly, the opposite effects of syntax and salience on the initial weight preference in Japanese have clear predictions regarding how the two factors would interact in a multi-factorial model. These multi-factorial analyses are popular in psycholinguistic and corpus linguistics studies which model processing phenomena using multiple linguistic and non-linguistic predictors (e.g., syntax, phonology, pragmatics, etc). The MG parsing models, in addition to offering explanatory accounts for various processing phenomena, highlights syntactic structure as a predicting factor in isolation, which helps put into context multi-factorial modeling results that are otherwise "difficult to calculate and even more difficult to interpret" (Gries, 2012, fn.11).

References

David Adger. 2003. *Core syntax: A minimalist approach*, volume 33. Oxford University Press Oxford.

John Frederick Bailyn. 2001. On scrambling: A reply to bošković and takahashi. *Linguistic inquiry*, 32(4):635–658.

Željko Bošković. 2004. Topicalization, focalization, lexical insertion, and scrambling. *Linguistic inquiry*, 35(4):613–638.

Željko Bošković and Daiko Takahashi. 1998. Scrambling and last resort. *Linguistic inquiry*, 29(3):347–366.

Bert Cappelle. 2005. Particle patterns in english: A comprehensive coverage.

Noam Chomsky. 1977. On wh-movement. *Formal Syntax*, pages 71–132.

Noam Chomsky. 2014. *The minimalist program*. MIT press.

Aniello De Santo. 2019. Testing a minimalist grammar parser on italian relative clause asymmetries. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 93–104.

Nicole Dehé. 2002. *Particle verbs in English: Syntax, information structure and intonation*, volume 59. John Benjamins Publishing.

Bruce Fraser. 1966. Some remarks on the ve rb-p art ic le construction in english. *Monograph Series on Languages and Linguistics*, page 45.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Thomas Graf, James Monette, and Chong Zhang. 2017. Relative clauses as a benchmark for minimalist parsing. *Journal of Language Modelling*, 5(1):57–106.

Stefan Gries. 2012. The influence of processing on syntactic variation: Particle placement in english. In *Verb-particle explorations*, pages 269–288. De Gruyter Mouton.

John A Hawkins. 1994. *A performance theory of order and constituency*, volume 73. Cambridge University Press.

Kyle Johnson. 1991. Object positions. *Natural Language & Linguistic Theory*, 9(4):577–636.

Aravind K Joshi. 1990. Processing crossed and nested dependencies: An automation perspective on the psycholinguistic results. *Language and cognitive processes*, 5(1):1–27.

Richard S Kayne. 1994. *The antisymmetry of syntax*. 25. mit Press.

Gregory M Kobele, Sabrina Gerth, and John Hale. 2013. Memory resource allocation in top-down minimalist parsing. In *Formal Grammar*, pages 32–51. Springer.

Richard K Larson. 1998. *Light predicate raising*. MIT. Center for Cognitive Science.

So Young Lee. 2018. A minimalist parsing account of attachment ambiguity in english and korean. *Journal of Cognitive Science*, 19(3):291–329.

Lei Liu. 2018. Minimalist parsing of heavy np shift. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*.

Lei Liu. 2022. *Phrasal Weight Effect on Word Order*. Ph.D. thesis, State University of New York at Stony Brook.

Barbara Lohse, John A Hawkins, and Thomas Wasow. 2004. Domain minimization in english verb-particle constructions. *Language*, pages 238–261.

Shigeru Miyagawa. 1997. Against optional scrambling. *Linguistic inquiry*, pages 1–25.

Robert Pasternak and Thomas Graf. 2021. Cyclic scope and processing difficulty in a minimalist parser. *Glossa: a journal of general linguistics*, 6(1).

Owen Rambow and Aravind K Joshi. 2015. 12 a processing model for free word-order languages. *Perspectives on sentence processing*.

John Robert Ross. 1967. Constraints on variables in syntax.

John Robert Ross. 1986. *Infinite syntax*. Ablex Publishing Corporation.

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

Edward Stabler. 1996. Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95. Springer.

Edward P Stabler. 2013. Two models of minimalist, incremental syntactic analysis. *Topics in cognitive science*, 5(3):611–633.

Lynne M Stallings and Maryellen C MacDonald. 2011. It's not just the "heavy np": relative phrase length modulates the production of heavy-np shift. *Journal of psycholinguistic research*, 40(3):177–187.

John Torr and Edward Stabler. 2016. Coordination in minimalist grammars: Excorporation and across the board (head) movement. In *Proceedings of the 12th international workshop on tree adjoining grammars and related formalisms (TAG+ 12)*, pages 1–17.

Thomas Wasow. 1997. Remarks on grammatical weight. *Language variation and change*, 9(1):81–105.

Thomas Wasow. 2002. *Postverbal behavior*. CSLI Stanford.

Hiroko Yamashita and Franklin Chang. 2001. "long before short" preference in the production of a head-final language. *Cognition*, 81(2):B45–B55.