Left-corner Minimalist parsing of mixed word order preferences

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

Abstract

This paper proposes a uniform, structure-based account for mixed word order preferences crosslinguistically. These preferences include the short-before-long preference in the English heavy NP shift, the long-before-short preference in the Japanese transitive sentences, and the absence of word order preference in Mandarin Chinese preverbal PPs. The syntactic structures of each competing word orders are formally characterized using Minimalist grammars (MGs) and constructed with a left-corner MG parser. Complexity metrics are derived from the parser's behavior, which relate the difficulties of the structure building process to memory load. The metrics show that the preferred word orders are less memory-intensive to build than their counterparts in both the short-before-long and the long-before-short cases, while no memory resource differences are found for the case where no word order preference exists. The results suggest that the preferred word orders – or a lack thereof – follow from their syntactic structures. This further supports the viability of left-corner MG parsing as a psycholinguistically adequate model for human sentence processing.

1 Introduction

Word preferences are conditioned by at least two factors: a general efficiency principle to minimize dependency length and language-specific syntactic characteristics. The efficiency principle reflects the tendency of grammars to minimize the dependency lengths between syntactic elements. This principle takes the form of Dependency Length Minimization (DLM, Hawkins 1994, 2004) when focusing on the lengths of syntactic dependency relations; and as the Dependency Locality Theory (DLT, Gibson 2000) when focusing on the memory resource required to hold those dependencies. Prior research has shown that this efficiency principle accounts for the short-before-long order in head-initial languages (e.g., Wasow, 2002) and the long-before-short preference in head-final languages (e.g., Hawkins, 1994)

The second factor conditioning word order preferences, language-specific syntactic characteristics, helps explain word preference variations across languages. For example, Liu (2020) notes that the association between headedness and word order preference does not always hold crosslinguistically. Other language-specific properties should therefore be considered in understanding word order preferences. These include the degree of word order flexibility, the prominence of NPs (Yamashita and Chang, 2001) and the richness of the case marking system (Futrell et al., 2020), all of which interact with broader structural tendencies to shape observed preferences.

Despite fruitful results and increasing empirical coverage of the research on the two factors, the interplay between the efficiency principle and language-specific syntactic characteristics remains puzzling. One key issue is that it is unclear what syntactic features and in what ways affect the preference for DLM. Research on DLM often relies on dependency grammar as the description of syntax and measures dependency length in terms of the number of intervening words. While this approach is simple and effective for large-scale corpus studies, it may overlook important syntactic information that contributes to word order preferences. For example, Liu (2008) argues that in a language such as Chinese, the richness of functional words might add extra distance to heads and their dependents when compared to a language such as English, where the grammatical functions are realized by inflection. This accounts for the larger mean dependency distance of Chinese. However, it remains unclear whether it is the additional morphemes themselves in Chinese, the different syntactic processes these functional heads undergo, or the syntactic structure they occupy, that contributes

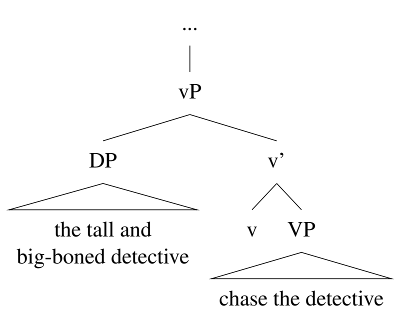
Proceedings of the Society for Computation in Linguistics (SCiL) 2025, pages 235-243.
Eugene, Oregon, July 18-20, 2025

to the dependency length difference.

This paper aims to address the interplay of the general efficiency principle and specific syntactic characteristics in predicting word order preferences from a Minimalist parsing perspective. Minimalist parsing is particularly well-suited for this task because its complexity metrics rigorously relate detailed syntactic structures to a general processing constraint: memory resources. Kobele et al. (2013) measure memory resources associated with a top-down MG parser using *tenure*, the amount of time a tree node is retained in memory. The authors argue that *tenure* can be viewed as a generalization of the DLT principles which correlates processing difficulties with memory space needed for holding dependency relations. They show that *tenure*-based complexity metrics are shown to successfully model processing contrasts between verb clusters in Dutch and German, and center and right embeddings in English. Recent work has expanded the empirical coverage of this MG processing modeling program (e.g., stacked relative clauses in Mandarin and English Zhang 2017; attachment ambiguity in English and Korean Lee 2018; gradient difficulty in Italian relative clauses De Santo 2019, 2020; end-weight preference in English and Mandarin Liu 2022, among others).

One limitation of the top-down MG processing model is that it encounters difficulty capturing the long-before-short preference in Japanese transitive sentence (Liu, 2022, 2023). Intuitively, word order preferences arise when speakers try to order long constituents around other shorter ones to ease processing. This shows up in syntactic trees as unbalanced sister nodes. For instance, in an English sentence *The tall and big-boned detective chased the suspect*, the subject and the vP is a pair of unbalanced sister nodes, as shown in (1).

(1)



When no other syntactic operations are involved, the top-down parser explores the structure top-down and from left to right to follow the word order. After the parser expands vP to DP and v', exploring either branch requires the parser to store the other branch in memory. This makes exploring the less complex branch more memory-efficient, which is the intuition behind the short-before-long preference. And in order to derive the opposite order preference in Japanese, additional structural assumptions are needed, which presents a challenge to the model (Liu, 2022, 2023).

Against this background, we opt for the left-corner parser for MGs in this study. We argue that the left-corner Minimalist parsing model effectively captures the short-before-long, the long-before-short preferences, and the absence of order preference. According to the modeling results, the preferred word orders require fewer memory resources to build than their counterparts. Furthermore, no memory load difference is found for structures that do not exhibit order preferences.

The remainder of the paper proceeds as follows. Section 2 introduces Minimalist Grammars (MGs), a left-corner MG parser, and the key complexity metrics for our parsing model. Section 3 presents modeling results of the three word order preferences. Section 4 concludes the paper with a discussion on the role of syntactic assumptions in the parsing model.

2 Left-corner Minimalist parsing

The left-corner Minimalist parsing approach to processing modeling consists of three components: characterizing syntactic proposals using Minimalist Grammars (MGs), incorporating the formalisms into left-corner parsing models, evaluating modeling results based on complexity metrics connecting parsing difficulty to memory load.

Minimalist Grammar is chosen as the formalism for two reasons. First, it incorporates the toolbox needed for Chomskyan syntax, providing detailed structural information known to influence processing. Second, MG parsers are available and relatively well-understood from previous studies (top-down MG parsing: Stabler 2013; Kobele et al. 2013, left-corner MG parsing: Stanojević and Stabler 2018; Hunter et al. 2019).

A left-corner MG parser is used instead of a top-down parser to overcome known difficulties of the latter as discussed above. The left-corner MG parser, on the other hand, has been recently argued to be a plausible model for human sentence processing (Liu, 2024).

The following subsections introduce the grammar formalism and its left-corner parser, and the key complexity metric needed for the subsequent modeling work.

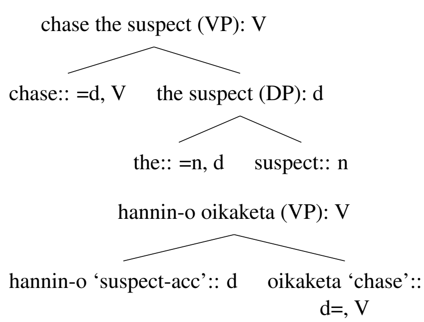
2.1 Minimalist Grammar and left-corner MG parser

Minimalist Grammar (MG, Stabler 1997, 2011) is a lexicalized, context-sensitive grammar formalism based on the Minimalist Program (Chomsky, 2014). In MGs, lexical items (LIs) are finite sequences of features containing information about sound, word shapes, and instructions for structure building operations. The grammar makes use of two such operations, `merge`, which combines categories, and `move`, which regulates movements.

`Merge` happens when two LIs have matching selector-selectee features as their first features. (2) illustrates how `Merge` builds a VP in English and Japanese.

(2) a.

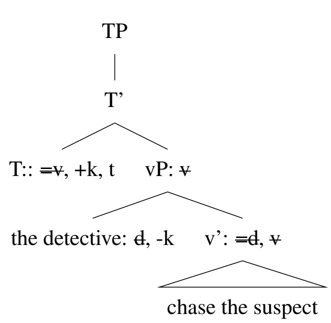
b.



To build the VP, the objects bear the same selectee feature `d` in both the English and the Japanese cases. The selector feature of the verb is `=d` in English and `d=` in Japanese. The placement of the equal sign (=) indicates the selectee to be merged on the left or the right. This allows our model to capture headedness.

`Move` happens when two LIs have matching licensor-licensee features as their first features, often written as polar pairs (e.g., +f, -f). This is illustrated in (3).

(3)



In (3), after other merge features are checked, the T head and the subject DP have matching `k` features as their first features. Movement is licensed. In contrast to a phrase structure tree where the mover is indicated at its landing site, the subject remains at its merge position in (3). Trees such as this are derivation trees. The central role derivation trees play in MGs and MG parsing is discussed in Graf et al. (2017). We will also use derivation trees as the data structure for our processing model.

A note on notation before proceeding. In the above derivation trees, double-colon (::) indicates a LI, while a single colon (:) indicates a derived category. Phrase node names are added wherever helpful for readability. For all subsequent trees, we will omit features, lexical/derived category distinctions, and use phrase names for tree nodes. Movement arrows will also be added when helpful.

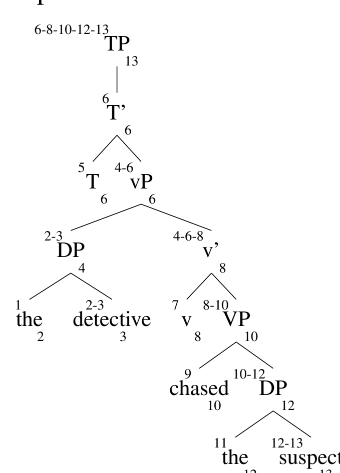
2.2 Left-corner MG parsing and complexity metrics

MG parsing can be viewed as a structural building process where a parser operates on MG rules, takes a string of words as input, and outputs a derivation tree when there is a valid parse. The left-corner parser for MGs used in this study is an arc-eager move-eager left-corner parser based on Stanojević and Stabler (2018); Hunter et al. (2019), in which the readers can find the full definitions of the parsing rules. For our purpose, we focus on tree annotations which are faithful visual representations of how the parser builds/traverses derivation trees.

Consider an arc-eager move-eager left-corner parse for the sentence (with silent nodes and string spans added) in (4). The parse history is represented using tree annotations in (5).

(4) 1 The 2 detective 3 T 3 v 3 chased 4 the 5 suspect 6

(5)



Following conventions in top-down MG parsing literature (e.g., Kobele et al. 2013; Graf et al. 2017), the superscripts and subscripts on the tree nodes, called indices and outdices, represent the steps at which that node enters and exits the memory storage of the parser. The dashes in the index of a node, which we use uniquely for left-corner parsing, connect the steps at which the parser updates its prediction regarding that node. Derivation trees annotated with indices, outdices, and dashes are shown to be condensed yet complete representations of the behavior of the left-corner MG parser (Liu, 2023, under revision). Building on this, we focus on the parser's updates represented with the dashes in the indices and show how to build complexity metrics based on them.

The update can be understood by examining the correspondence between parse items and derivation tree fragments. One node in the derivation tree can correspond to multiple strictly different parse items for a left-corner MG parse. For example, in (5) the parser reads the first input word *the* (step 1) and makes a left-corner prediction based on it (step 2), creating a parse item which takes the form of an implication shown in (6).

(6) `(2-i) n, M => (1-i) d, M`

This parse item is interpreted as follows, if from the string span of `(2-i)` the parser finds an item with category feature `n` and an optional mover chain `M`, the parser can infer that from the string span of `(1-i)` there is an item of category `d` which carries over the mover chain `M`. In terms of tree fragments, (6) corresponds to a DP with a daughter node yet to be confirmed. This is also the tree portion annotated with indices and outdices up to 2, matching the steps so far.

Next, when the parser reads *detective* from the input (step 3), the left-hand side of the implication in (6) is satisfied, a new parse item (7) is created at the same step and replaces (6).

(7) `(1-2) d`

This parse item means that from the string span of `(1-2)`, there is an item of category `d` without any mover chain. In terms of tree fragments, (7) corresponds to the fully built DP *the detective*. At step 3, both daughters of the DP are fully annotated. The DP node itself has an index of 3 and no outdex, meaning that it is still in memory at this step, ready for further operations.

Both the right-hand side in (6) and the whole item in (7) correspond to the same DP node in the derivation tree. The parser updates its knowledge of the node from a conditioned inference to a confirmed node. And the dashed index on the DP node records the steps at which the parser makes those updates. By taking the difference between the two dash-connected steps, we get the number of steps a parse item needs to be stored in memory, or its *item tenure*. For example, the parse item in (6) has a trivial item tenure of 1, as it is only stored between steps 2 and 3.

For a non-trivial example, vP has in its index 4-6. The parser first updates its knowledge on the vP node when it makes a left-corner prediction based on the DP *the detective*. A vP with a daughter node yet to be confirmed is created and held in memory. The parser's second update happens after the T head is read and processed. The time between the two updates is recorded with the dash-connected step pair. By taking the difference of the pair, we have the item tenure of the partially built vP, 2.

Item tenure serves as the basis for the complexity metrics of our left-corner MG parsing model. There are many ways to construct complexity metrics based on item tenure. Liu (under revision) explores a few of those possibilities. Here we focus on Maximal item tenure (MaxT_{item}) and its recursive variant (MaxT^{R}_{item}). MaxT_{item} is the maximal duration that any parse item remains in memory. MaxT^{R}_{item}, following Graf et al. (2017), applies MaxT_{item} recursively. MaxT_{item} is shown to be able to capture the processing of sentence embeddings (Liu, 2024), it is included here to further test its reliability. In cases of a lack of word order preferences, we expect to find a tie in MaxT_{item} for the word order pair. Examining MaxT^{R}_{item} in those cases helps reveal further potential processing differences.

With methods and tools ready, we turn to the modeling results.

3 Modeling results

The processing phenomena modeled with the left-corner MG parser are the short-before-long preference in the English heavy NP shift (HNPS); the long-before-short preference in the Japanese transitive sentences; and the absence of word order preference in preverbal PPs in Mandarin Chinese. For each case, we make pairwise comparisons between the two opposite word orders (e.g., shift vs. canonical word order in English heavy NP sentences).

Overall, MaxT_{item} successfully captures all three word order preferences. The preferred order has a lower MaxT_{item} in both the English (short-before-long) and Japanese (long-before-short) target sentences. Furthermore, MaxT_{item} predicts a tie in processing difficulties in the Mandarin (no preference) sentences. Since our goal is to understand the interplay of specific syntactic structures and a general memory constraint on processing, we next examine the structural assumptions and the complexity metric in each word order pair.

3.1 Short-before-long preference

The target sentences for the short-before-long preference are the canonical (8) and heavy NP shift order (9) in English (with silent heads).

(8) Max T v-put all the boxes of home furnishings V in a car.

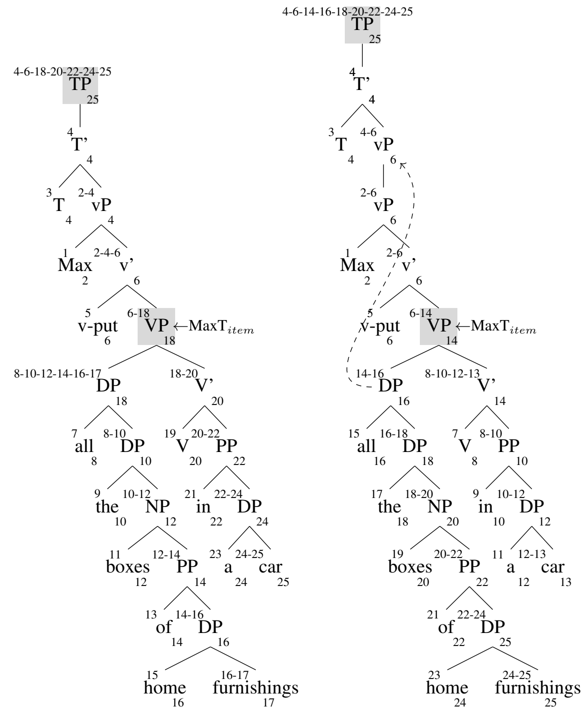
(9) Max T v-put V in a car all the boxes of home furnishings.

Evidence for the short-before-long preference in the above sentences is found in numerous behavioral and corpus studies (e.g., behavioral: Stallings et al. 1998; Stallings and MacDonald 2011; corpus: Wasow 2002; Liu 2020). For our model, we expect to find that the shifted order has a lower MaxT_{item} compared with that of the canonical order, suggesting that the former is easier to process.

In terms of structural assumptions, a rightward movement analysis (Ross, 1986; Overfelt, 2015) is adopted to derive the heavy NP shift order. V-to-v and AgrO movements are factored out for simplicity.

The modeling results suggest that the shift order is easier to process than the canonical order. MaxT_{item} for the shift order is 12 compared with 8 for the canonical order. The reason for the difference in MaxT_{item} can be seen from the tree annotations in Figure 1.

For both word orders, the MaxT_{item} is associated with the VP node. As the parser processes the verb *v-put*, a left-corner prediction based on the node predicts and stores an implicational parse item involving VP: if the parser finds a VP, it can confirm that there is a TP. Given the arc-eager strategy, this stored VP node is considered found when the parser makes a left-corner prediction based on one of its fully built daughter. And this is when word order makes a difference. If the parser first builds the less complex daughter, the V', the VP is held in memory for less time than when building



(a) HNPS - Canonical order (b) HNPS - Shift order

Figure 1: Tree annotations for short-before-long preference

the more complex daughter first. This is reflected in the difference in MaxT_{item}, as can be seen in Figure 1a for the canonical order and Figure 1b for the shift order.

This is an encouraging result as it indicates that the left-corner MG parsing is at least as good as its top-down variant in capturing the short-before-long preference. We now turn to the long-before-short preference, where the top-down model struggles.

3.2 Long-before-short preference

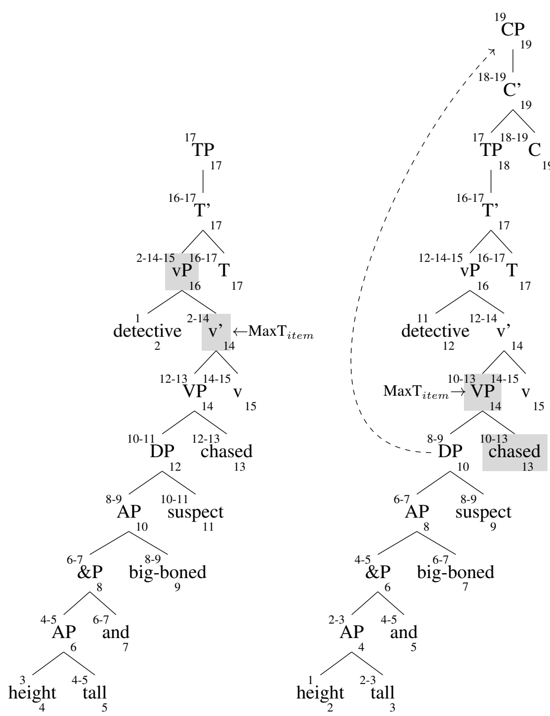
The long-before-short preference we model is reported in Yamashita and Chang (2001) regarding Japanese transitive sentences. The study finds that in a sentence production task, Japanese-speaking participants tend to order long arguments ahead of short ones. For example, compared with a canonical SOV order in (10), a long-before-short OSV order in (11) is preferred when the object is long.

(10) keezi-ga Se-ga takakute
detective-nom height-nom tall-and
gassiri sita hanni-o oikaketa v T
big-boned suspect-acc chased

(11) Se-ga takakute gassiri sita hanni-o
height-nom tall-and big-boned suspect-acc
keezi-ga oikaketa v T C
detective-nom chased
'The detective chased the suspect who is tall and big-boned.'
(adapted from Yamashita and Chang 2001, silent nodes added)

The sentence pair in (10-11) is used in our model as target sentences. A scrambling analysis is assumed to derive the long-before-short order (Saito, 1992). V-to-v and AgrO movements are again factored out for simplicity.

The modeling results show that the shift, long-before-short word order is easier to process than the canonical order. MaxT_{item} of the shift order is 3 compared with 12 of the canonical order. The tree annotations confirm the processing prediction.



(a) Japanese - SOV order (b) Japanese - OSV order

Figure 2: Tree annotations for long-before-short preference

In Figure 2a which corresponds to the canonical order, MaxT_{item} is associated with the v' node. The parser predicts and stores a parse item with v' when the subject, *detective*, is processed. The parse item is flushed from memory when one of the daughters of v' is built and used for left-corner prediction. Given the word order, this only happens after the long DP (indeed, the full VP) is fully built, resulting in large item tenure. In the long-before-short tree in Figure 2b, the parser builds the long DP first, during which process no other parse item is held in memory. As a result, item tenures and MaxT_{item} stay relatively low throughout the parse, predicting that the long-before-short order is easier to process than the canonical order.

3.3 Absence of order preference

Liu (2020) reports in a large-scale corpus that Mandarin Chinese preverbal PPs lack a preference for word order when the two PPs are of different lengths. For example, no word order preference is found between whether ordering the longer PP first (12) or the shorter first (13).

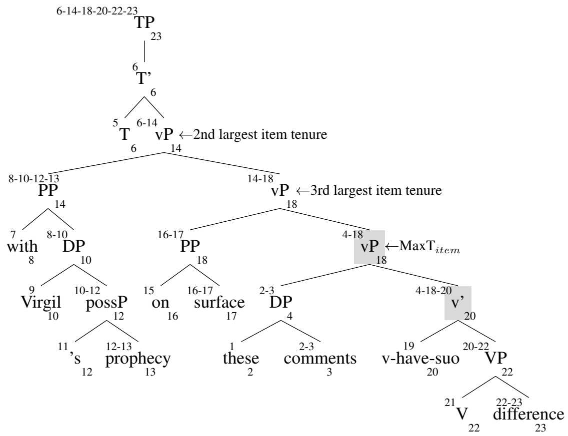
(12) zhexie yanlun T [he weijier de yuyan]
these comments with Virgil's prophecy
[zai biaomian] v-you-suo V churu
on the surface have-suo differences

(13) zhexie yanlun T [zai biaomian]
these comments on the surface
[he weijier de yuyan] v-you-suo V
with Virgil's prophecy have-suo
churu
differences
'These comments have differences on the surface with Virgil's prophecy.'
(from Liu 2020, silent nodes added)

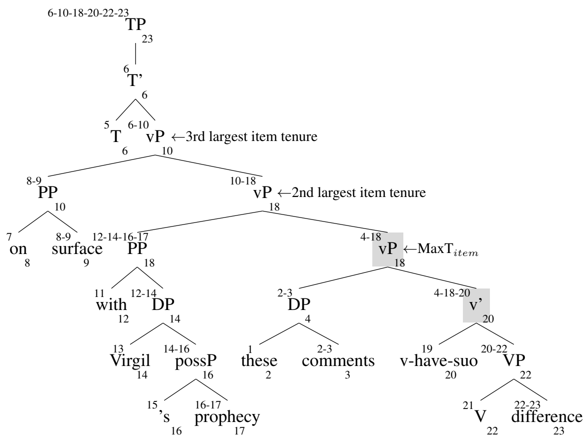
(12) and (13) are the target sentences to include in our model. In terms of the structural assumption, the two PPs are considered based-generated adjuncts. Similar to before, V-to-v and AgrO movements are factored out for simplicity. Unlike before, the two word orders are not derivationally related under the current structural assumption. We will consider an alternative analysis in the context of methodological discussion in Section 4.

The results show that the two orders are indistinguishable for our model based on MaxT_{item}. MaxT_{item} is 14 for both orders, suggesting that no preference is expected for the two word orders. We see why MaxT_{item} is unaffected by word order alternations in the tree annotations in Figure 3.

Given the current structural assumption, MaxT_{item} is associated with the vP node immediately dominates the subject *these comments*. The parser creates and stores a parse item with this vP node when the subject is processed. This parse item is flushed from memory after the inner PP, or the linearly second PP, is processed. Alternating the order of the two PPs would not affect the item



(a) Mandarin Chinese - long PP first



(b) Mandarin Chinese - short PP first

Figure 3: Tree annotations for Mandarin Preverbal PPs

tenure of the parse item with the vP node created early on.

Interestingly, MaxT^R_{item}, a recursive evaluation of MaxT_{item}, also predicts that there is no preference between the two orders. In the two orders, the second largest item tenures are equal, so are the third largest. They are associated with the mother node of the longer and the short PPs respectively. Because of the structural similarity, all other item tenures are equal, too. An alternation of word order does not affect the item tenure profile.

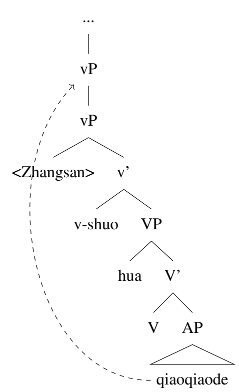
4 Discussions: an alternative structure for Mandarin adjuncts

The modeling results have shown that left-corner MG parsing is an effective model for word order preferences crosslinguistically. MaxT_{item} has proven to be a reliable complexity metric capturing the mixed word order preferences under the current syntactic assumptions. Among those assumptions, the base-generation analysis of Mandarin preverbal PPs warrants particular attention. While it is standard to treat PP adjunction as base-generation, with word order alternation derived from different base merge positions, the choice of this structural assumption has a potential limitation: it can be adequately captured by a Context-Free Grammar. For both formalisms, no movement is involved that causes a mismatch between the string order and the leaf order. The ability to handle this mismatch distinguishes MG parsers from CFG parsers (Graf et al., 2017). As a result, for our purposes, processing models based on this syntactic assumption may not fully highlight the unique contribution of MG parsing in capturing the interplay between general efficiency principles and detailed syntactic structures.

Furthermore, there are syntactic proposals regarding other types of adjuncts in Mandarin that require the expressive power of MGs. For example, (Larson, 2018) argues that manner adverbs in Mandarin Chinese merge as VP complement and move to vP edge which derives the correct word order. This is schematized in (14).

(14) a. Zhangsan qiaoqiaode shuo hua
Z quiet-de speak words
'Z. speaks quietly.' (Larson, 2018)

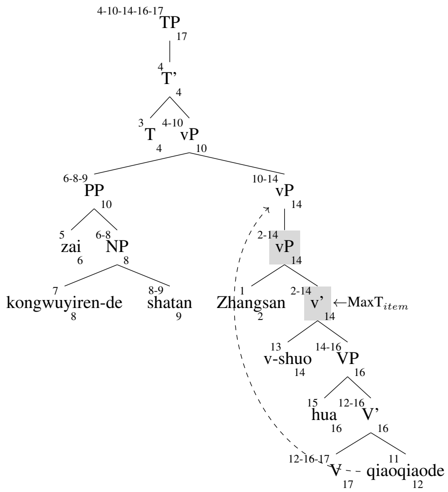
b.



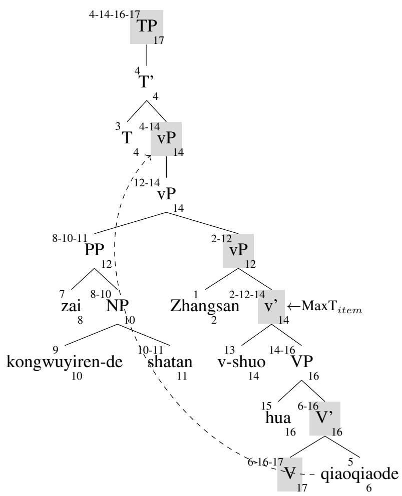
We next model how this syntactic proposal affects order preferences. The target sentences (with silent heads) are shown in (15) and (16) corresponding to the PP-first and adverb-first order, respectively.

(15) Zhangsan T zai kongwuyiren de shatan
Z. at not-a-single-person de beach
qiaoqiaode v-shuo hua V
quite-de speak word

(16) Zhangsan T qiaoqiaode zai
Z. quite-de at
kongwuyiren de shatan v-shuo hua V
not-a-single-person de beach speak word



(a) Mandarin Chinese - PP first



(b) Mandarin Chinese - adverb first

Figure 4: Tree annotations for Mandarin PP and AP adjuncts

'Z. speaks quietly at an empty beach.'

For syntactic assumptions, the manner adverb is analyzed according to Larson (2018). The PP adjunct is base-generated either before or after the manner adverb moves to derive the two word orders. This is illustrated with annotated derivation trees in Figure 4.

The modeling result suggests that an AP-first order is preferred irrespective of the length of the two phrases. In both word orders, MaxT_{item} is associated with the mother and sister node of the subject *Zhangsan*. The parse item associated with the two nodes is stored until the parser updates its knowledge on either node. For both orders, this happens after the parser has processed the AP and the PP. This means the lengths of the two phrases have the same effect on MaxT_{item} for both orders. In the PP-first case in Figure 4a, it is the v' node that gets an update as the parser processes the two adjuncts and the verb *v-shuo*. In the AP-first case in Figure 4b, the vP node gets an update as soon as the two adjuncts are built and processed. This results in a constant MaxT_{item} advantage of 2 (10 vs. 12) for the AP-first order over the PP-first order.

The result does not immediately rule out the possibility that there is no preference for ordering shorter or longer phrases first. Empirical data is needed to verify whether there is a preference for AP-first ordering and to assess its implications for the DLM principle. We leave these intriguing questions for future research.

5 Conclusion

This paper offered a unified, structure-based account of crosslinguistic word order preferences using Minimalist Grammars and a left-corner MG parser. The results show that preferred word orders correspond to structures that are less memory-intensive to process, and that no memory load difference is observed—given the current complexity metric—in cases that lack a word order preference. This supports the view that word order preferences follow from syntactic structure and highlights the potential of left-corner MG parsing as a psycholinguistically grounded model of sentence processing.

References

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.

Aniello De Santo. 2020. *Structure and memory: A computational model of storage, gradience, and priming*. Ph.D. thesis, State University of New York at Stony Brook.

Richard Futrell, Roger P Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96(2):371–412.

Edward Gibson. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, 2000:95–126.

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

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

John A Hawkins. 2004. *Efficiency and complexity in grammars*. Oxford University Press on Demand.

Tim Hunter, Miloš Stanojević, and Edward Stabler. 2019. The active-filler strategy in a move-eager left-corner minimalist grammar parser. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–10.

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. 2018. Ap-de adverbs in mandarin. *Studies in Chinese Linguistics*, 39(1):1–28.

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

Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191.

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

Lei Liu. 2023. Processing advantages of end-weight. *Proceedings of the Society for Computation in Linguistics*, 6(1):250–258.

Lei Liu. 2024. Psycholinguistic adequacy of left-corner parsing for minimalist grammars. *Proceedings of the Society for Computation in Linguistics (SCiL)*, pages 275–280.

Lei Liu. under revision. Psycholinguistic plausibility of left-corner parsing with Minimalist Grammars.

Zoey Liu. 2020. Mixed evidence for crosslinguistic dependency length minimization. *STUF-Language Typology and Universals*, 73(4):605–633.

Jason Overfelt. 2015. Rightward movement: A study in locality.

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. 1997. Derivational minimalism. In *Logical Aspects of Computational Linguistics: First International Conference, LACL'96, Nancy, France, September 23-25, 1996. Selected Papers*, volume 1328, page 68. Springer Science & Business Media.

Edward P Stabler. 2011. Computational perspectives on minimalism. *Oxford handbook of linguistic minimalism*, pages 617–643.

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.

Lynne M Stallings, Maryellen C MacDonald, and Padraig G O'Seaghdha. 1998. Phrasal ordering constraints in sentence production: Phrase length and verb disposition in heavy-np shift. *Journal of Memory and Language*, 39(3):392–417.

Miloš Stanojević and Edward Stabler. 2018. A sound and complete left-corner parsing for minimalist grammars. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 65–74.

T Wasow. 2002. *Postverbal Behavior*. CSLI Lecture Notes (CSLI- CHUP) Series. CSLI Publications.

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

Chong Zhang. 2017. *Stacked Relatives: Their Structure, Processing and Computation*. Ph.D. thesis, State University of New York at Stony Brook.