Preface: SCiL 2023 Editors' Note

Tim Hunter[1] and Brandon Prickett[2]
[1]University of California, Los Angeles, [2]University of Massachusetts Amherst

This volume contains research presented at the sixth annual meeting of the Society for Computation in Linguistics (SCiL), held in Amherst, Massachusetts, June 15–17, 2023.

Research was submitted to be reviewed either in the form of a paper, or as an abstract. The oral presentations, or talks, at the conference included both papers and abstracts. Authors of accepted abstracts were given the option of publishing an extended version; these are included with the papers in this volume.

In total, we received 68 submissions to the conference, 28 abstracts and 40 papers. 19 submissions were selected for oral presentation (~28%) and 30 for poster presentation (~44%).

We thank our reviewers for their indispensable help in selecting the research for presentation at the conference:

> Adam Jardine, Alexander Clark, Alicia Parrish, Andrew Lamont, Aniello De Santo, Bonnie L. Webber, Brian Dillon, Caitlin Smith, Canaan Breiss, Carolyn Jane Anderson, Christo Kirov, Christopher Potts, Connor Mayer, D. Terence Langendoen, Dongsung Kim, Dylan Bumford, Edward P. Stabler, Emily Morgan, Eric Raimy, Eric Rosen, Giorgio Magri, Gregory M. Kobele, Hossep Dolatian, Itamar Kastner, Joe Pater, Jonathan Brennan, Jordan Kodner, Katrin Erk, Laurel Perkins, Lindy Comstock, Lucy Li, Marina Ermolaeva, Matthew Goldrick, Nathan Schneider, Nick Huang, Olga Zamaraeva, Philippe de Groote, Qihui Xu, Robert Frank, Robert Malouf, Rui Chaves, Sebastian Schuster, Sheng-Fu Wang, Tal Linzen, Thomas Graf, Tiago Pimentel, Timothee Mickus, Tracy Holloway King, William Idsardi, and Yohei Oseki.

Thanks also to Joe Pater and Steve Bischof for logistical help.

SCiL 2023 also included invited talks by Jane Chandlee (Haverford College), Brendan O'Connor (University of Massachusetts Amherst), and Emily Morgan (University of California, Davis). Further information can be found at our website: https://blogs.umass.edu/scil/.

A third structure building operation for Minimalist Grammars

Johannes Schneider
Universität Leipzig
johannes.schneider@uni-leipzig.de

Abstract

I propose a new structure-building operation for Minimalist Grammars (Stabler 1997) which allows the grammar formalism to grow trees with more than one root. I demonstrate that together with the assumption that this new long-distance dependency holds between nominal arguments and their selectors, one can generate Horn amalgams and parasitic gaps with a number of desired properties.

1 Introduction

I propose a new structure-building operation *3rd-merge*, formalized within the framework of Minimalist Grammars (Stabler 1997), that makes it possible to generate tree structures with more than one root where the two roots are structurally independent except at a single connecting phrase, the pivot. Within minimalist syntax, a number of proposals exists to extend the formalism beyond the operations *merge* and *move*, most notably sidewards movement (Nunes 1995), parallel merge (Citko 2005) and grafts (van Riemsdijk 2006, van Riemsdijk 2010). The effect of *3rd-merge* is to allow the selector to long-distance select its argument out of an otherwise structurally independent root (I discuss similarities and differences to the above-mentioned extensions below). I propose that this new long-distance dependency underlies the phenomena of Horn amalgams and parasitic gaps.

Horn amalgams are constructions where two apparently independent clauses share a common element (Lakoff 1974):

(1) Joscha adores [I think it was cats].

cats appears to be both the argument of *adore* and *was*. The clause containing the latter verb is structurally independent from the matrix clause; *adore* does not select for the parenthesis-like clause but the noun *cats*. Neither of those clauses c-command the other. Evidence for this comes e.g. from the fact that binding (or any other syntactic operation) between elements from each clause is impossible (see Kluck 2011, ch.3 for an overview). The so-called pivot *cats* is therefore shared by two otherwise independent clauses and is the only element accessible to both clauses.

In parasitic gaps, an otherwise ungrammatical long-distance dependency (here: extraction out of an adjunct) becomes grammatical in certain configurations in the presence of a licit long-distance dependency (Engdahl 1983 i.a.):

(2) [Which article]$_1$ did you file t_1 [without reading pg_1]?

Both 'real' and parasitic gap refer to the same element. I argue that the matrix clause and the adjunct share the single element *which article* in the same manner as amalgams; the crucial difference is that the adjunct as additional root is reintroduced into the matrix root. When the pivot is moved, this creates the appearance of two gaps.

The structure of this article is as follows: I present the algebraic definition of Minimalist Grammars from Stabler and Keenan (2003) (Section 2). I then introduce the new operation and the rules describing its behaviour (Section 3), together with an application to the phenomena they are supposed to derive. Section 4 concludes with a comparison with other operations and a discussion of open issues.

2 Minimalist Grammars

Stabler and Keenan (2003) provide an algebraic definition of Minimalist Grammars (MGs). A Minimalist Grammar $G = \langle \Sigma, F, Types, Lex, \mathscr{F} \rangle$, with a non-empty alphabet Σ, the set of Features F consisting of *base* features (n,v,c,...), the respective selection features, as well as licensor and licensee features for movement, i.e. $F = base \cup \{=f | f \in$

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 6-14.
Amherst, Massachusetts, June 15-17, 2023

$base\} \cup \{+f | f \in base\} \cup \{-f | f \in base\}$[1], the $Types = \{::, :\}$, with $\cdot \in \{::, :\}$ as shorthand. They call Chains $C = \Sigma^* \times Types \times F^*$, and Expressions $E = C^+$, with the lexicon $Lex \subseteq C^+$ as a finite subset of $\Sigma^* \times \{::\} \times F^*$. There are two operations $\mathscr{F} = \{merge, move\}$, defined as partial functions as in Figure 1. I use $s, t, u, v, w \in \Sigma^*$; $\beta, \gamma \in F^*$; $\delta, \varepsilon, \zeta \in F^+$, chains $\alpha_1, ..., \alpha_k$ or $\iota_1, ..., \iota_\ell$ or $\mu_1, ..., \mu_m$ with $k, l, m \geq 0$.

$merge : (E \times E) \rightarrow E$ consists of three subcases, defining merge into complement position (*merge1*) and specifier position (*merge2*) and merge of a moving element (*merge3*). $move : E \rightarrow E$ is described by two functions for which the Shortest Move Constraint (SMC) holds: no member of $\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_k$ has $-f$ as first feature. There is either movement to a final (*move1*) or non-final (*move2*) landing site.

3 A new operation

The core intuition behind the new operation is to create a new type of long-distance dependency between nominal arguments and their selectors such as verbs that allows verbs to select an element from within an independent root. Crucially, the secondary root remains structurally independent except at the selected phrase. A possible visualization would be that of a form of 'long-distance' in-situ merge. I introduce a new type of positive/negative feature pair for the new operation *3rd-merge*: $\#f$ and $\nmid f$[2]. By assumption, there is only a single feature of this type in a language ($\nmid n$ or $\nmid d$, depending on what the assumed highest projection in the nominal domain of that language is). This restriction is driven purely by empirical considerations, to restrict the phenomena of amalgams and parasitic gaps to nominals (for now).

In addition to the symbols in the standard MG definition, I use $\psi \in F^3$ where $F^3 = \{\nmid f | f \in base\}$; ω is of the form $[t : \psi\gamma, \mu_1, ..., \mu_m]$ and $\varsigma \in \{n, c\}$. For the present purposes, I restrict the structure of potential lexical items as follows: $\{=f, \#f\}^*.f.(\nmid f.)-f^*$, i.e. there is at most one $\nmid f$ directly after the category symbol of any given lexical item; $\#f$ behaves like selector features. Figure 2 provides an overview of all rules.

Let us assume that the highest projection in the nominal domain is *n*, and that all nouns have both n and $\nmid n$ in their feature string ($cat :: n.\nmid n$). A consequence is that additional roots can only grow on top of nominals. The assumption is that nominals are always merged via an application of *3rd-merge*. Phrases with feature string $f.\nmid f$ are *3rd-merged* into complement or specifier position (*3merge-1/2*) or as moving item (*3merge-3*). Nominals are therefore treated as a trivial case of an independent root, namely one where no additional structure has grown on top of it. The category of this trivial root n is treated as syntactically inert after the application of *3rd-merge*.

The system also allows a non-trivial root to grow on top of a nominal before it is *3rd-merged*. I call such a root the *secondary root*[3] (e.g. the bracketed '*I think it was*' in (1)) since *3rd-merge* creates an asymmetry between the roots, as will be discussed below.

The rule *merge4* governs the special case where growth of a non-trivial secondary root is initiated on top of a nominal. A head selects for a category feature of an expression that is followed by $\nmid f$. The merge features of the argument are erased but it becomes part of the chains of the selector, akin to merging moving expressions, and becomes inaccessible for the rest of the derivation within the secondary root until it is selected via an application of *3rd-merge* out of a different root. The inaccessible pivot is indicated by square brackets. I denote bracketed elements of the form $[t : \psi\gamma, \mu_1, ..., \mu_m]$ as ω. Note that so far this operation is only defined to apply in complement position. Example derivations for an amalgam and a parasitic gap can be found in Figure 3. '*I think it was*' and '*without reading*' are treated as such secondary roots, and the first steps in both derivations (selection by *was* or *reading*) exemplifies an application of *merge4*.

The introduction of ω by *merge4* now requires an update to the former *merge* and *move* rules so that an expression can contain 0 or 1 ω ('ω' abbreviates '0 or 1 ω' for readability). *merge1* and *merge2* remain unaffected save a potential presence of an inert ω. The argument in *merge1* and the function in *merge2* can contain ω. Note that, similar to the complement-only restriction for *merge4*, I disallow merging an expression into specifier position that contains ω. This would allow a potentially unbounded number of ω in an expression, with potentially non-trivial nesting, something I want to

[1] Note that there is some redundancy in this definition since there are no base movement features as categories.

[2] 'plus-equals' and 'minus-equals', for lack of better terms.

[3] I use 'root' here as pars pro toto for the whole single-rooted subtree in a multi-rooted tree.

$$\frac{s :: =f\delta \qquad t \cdot f, \alpha_1, ..., \alpha_k}{st : \delta, \alpha_1, ..., \alpha_k}\ \text{merge1} \qquad \frac{s \cdot =f\delta, \alpha_1, ..., \alpha_k \qquad t \cdot f\varepsilon, \iota_1, ..., \iota_\ell}{s : \delta, \alpha_1, ..., \alpha_k, t : \varepsilon, \iota_1, ..., \iota_\ell}\ \text{merge3}$$

$$\frac{s : =f\delta, \alpha_1, ..., \alpha_k \qquad t \cdot f, \iota_1, ..., \iota_\ell}{ts : \delta, \alpha_1, ..., \alpha_k, \iota_1, ..., \iota_\ell}\ \text{merge2} \qquad \frac{s : +f\delta, \alpha_1, ..., \alpha_{i-1}, t : -f, \alpha_{i+1}, ..., \alpha_k}{ts : \delta, \alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_k}\ \text{move1}$$

$$\frac{s : +f\delta, \alpha_1, ..., \alpha_{i-1}, t : -f\varepsilon, \alpha_{i+1}, ..., \alpha_k}{s : \delta, \alpha_1, ..., \alpha_{i-1}, t : \varepsilon, \alpha_{i+1}, ..., \alpha_k}\ \text{move2}$$

Figure 1: Standard MG rules

$$\frac{s :: =f\delta \qquad t \cdot f, \omega, \alpha_1, ..., \alpha_k}{st : \delta, \omega, \alpha_1, ..., \alpha_k}\ \text{merge1}$$

$$\frac{s : =f\delta, \omega, \alpha_1, ..., \alpha_k \qquad t \cdot f, \iota_1, ..., \iota_\ell}{ts : \delta, \omega, \alpha_1, ..., \alpha_k, \iota_1, ..., \iota_\ell}\ \text{merge2}$$

$$\frac{s : =f\delta, \omega, \alpha_1, ..., \alpha_k \qquad t \cdot f\varepsilon, \mu_1, ..., \mu_m}{s : \delta, \omega, \alpha_1, ..., \alpha_k, t : \varepsilon, \mu_1, ..., \mu_m}\ \text{merge3 spec}$$

$$\frac{s :: =f\delta \qquad t \cdot f\psi\gamma, \alpha_1, ..., \alpha_k}{s : \delta, [t : \psi\gamma, \alpha_1, ..., \alpha_k]}\ \text{merge4}$$

$$\frac{s : +f\delta, \omega, \alpha_1, ..., \alpha_{i-1}, u : -f, \alpha_{i+1}, ..., \alpha_k}{us : \delta, \omega, \alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_k}\ \text{move1}$$

$$\frac{s : +f\delta, \omega, \alpha_1, ..., \alpha_{i-1}, u : -f\varepsilon, \alpha_{i+1}, ..., \alpha_k}{s : \delta, \omega, \alpha_1, ..., \alpha_{i-1}, u : \varepsilon, \alpha_{i+1}, ..., \alpha_k}\ \text{move2}$$

$$\frac{s :: \#f\delta \qquad t \cdot f.\sharp f, \alpha_1, ..., \alpha_k}{st : \delta, \alpha_1, ..., \alpha_k}\ \text{3merge-1}$$

$$\frac{s : \#f\delta, \omega, \alpha_1, ..., \alpha_k \qquad t \cdot f.\sharp f, \iota_1, ..., \iota_\ell}{ts : \delta, \omega, \alpha_1, ..., \alpha_k, \iota_1, ..., \iota_\ell}\ \text{3merge-2}$$

$$\frac{s \cdot \#f\delta, \omega, \alpha_1, ..., \alpha_k \qquad t \cdot f.\sharp f\varepsilon, \iota_1, ..., \iota_\ell}{s : \delta, \omega, \alpha_1, ..., \alpha_k, t : \varepsilon, \iota_1, ..., \iota_\ell}\ \text{3merge-3}$$

$$\frac{s :: \#f\delta \qquad t : c, [u : \sharp f, \alpha_1, ..., \alpha_k]}{stu : \delta, \alpha_1, ..., \alpha_k}\ \text{3merge-1'}$$

$$\frac{s : \#f\delta, \omega, \alpha_1, ..., \alpha_k \qquad t : c, [u : \sharp f, \iota_1, ..., \iota_\ell]}{tus : \delta, \omega, \alpha_1, ..., \alpha_k, \iota_1, ..., \iota_\ell}\ \text{3merge-2'}$$

$$\frac{s \cdot \#f\delta, \omega, \alpha_1, ..., \alpha_k \qquad t : \varsigma\gamma, [u : \sharp f\varepsilon, \iota_1, ..., \iota_\ell]}{s : \delta, \omega, [t : \varsigma\gamma, u : \varepsilon], \iota_1, ..., \iota_\ell, \alpha_1, ..., \alpha_k}\ \text{3merge-4}$$

$$\frac{s : =\varsigma\delta, \omega, [t : \varsigma, u : \varepsilon], \alpha_1, ..., \alpha_k}{ts : \delta, \omega, \alpha_1, ..., \alpha_k, u : \varepsilon}\ \text{chain-merge1}$$

$$\frac{s : =\varsigma\delta, \omega, [t : \varsigma\varepsilon, u : \zeta], \alpha_1, ..., \alpha_k}{s : \delta, \omega, \alpha_1, ..., \alpha_k, t : \varepsilon, u : \zeta}\ \text{chain-merge2}$$

$$\frac{s : +f\delta, \omega, \alpha_1, ..., \alpha_{i-1}, [t : \varsigma\gamma, u : -f\varepsilon], \alpha_{i+1}, ..., \alpha_k}{s : \delta, \omega, \alpha_1, ..., \alpha_{i-1}, [t : \varsigma\gamma, u : \varepsilon], \alpha_{i+1}, ..., \alpha_k}\ \text{move3}$$

Figure 2: MGs with *3rd-merge*

exclude. There is thus a **ban on $\nmid f$ in specifiers.**

The original *merge3* is joined by the additional *merge3 spec* that allows for the possibility of a selector containing ω when merging into specifier position. A selector cannot contain ω when merging its complement, but I also exclude the option that its argument contains ω in this case. Such a rule would be 'unphysical' in that it would lead to 'root-distributed remnant movement'. An example would be VP-movement inside an amalgam minus its inert object as ω which appears in its in-situ position in the matrix root. To the best of my knowledge there is no such phenomenon. Previous *move* rules, however, are updated trivially to allow for the presence of inaccessible ω. In sum, *merge4* initiates growth of a secondary root, but previous *merge* and *move* rules continue to build structure in the familiar way, largely unaffected by the presence/absence of a pivot.

3.1 Amalgams

We now turn to the cases where *3rd-merge* selects an argument from within a non-trivial secondary root, as in Horn amalgams such as (1) where '*I think it was cats*' grows on top of the cordoned off NP *cats* with a $\nmid n$ feature. This clause, and the clause headed by *adore* are structurally independent. As mentioned, no element from either clause can bind an element from the other, i.e. there is no c-command relationship between these clauses[4]. Only the pivot is accessible for both clauses. In *3merge-1'/2'*, the selector selects (and therefore c-commands) only the element carrying $\nmid f$ from the bracketed chain, but does not establish any syntactic relationship with the rest of the expression. This property of the rule effectively introduces multiply-rooted trees since there is an undominated complete root tree with a single position inside where it can 'dock' with another root.

The rule enforces that the expression from which the argument of the selector originates is a complete CP which has no features unchecked besides the pivot in the bracketed part. The *c* on the head of the secondary root is *not* selected and remains unchecked but vanishes in the resulting expression. This implements the idea that a syntactic object is complete if the only unchecked features in *all* its roots are of start category *c*. Another effect that this rule enforces is a derivational 'timing' in that secondary roots can only be connected with a primary root (by definition the root whose head carries $\#f$) after they are built completely, but not in an intermediate stage. This would yield an expression with multiple heads that need to check their features, with often non-trivial bracketing. The rules above avoid this complication.

In the result of the rule, amalgam plus pivot are linearized as a single unit with respect to the verb, yielding the correct *adores I think it was cats*. This step is also illustrated in Figure 3. Also note that I do allow completed amalgams in specifier position (*3merge-2'*), it is only *unchecked* $\nmid f$-features that are disallowed, for the reasons discussed above.

As a last note, the rules allow for subextraction from pivots into the matrix root, as indicated by the presence of chains. This is empirically justified:

(3) Of which person does Daniel have [I think it was a painting *t*]?

There is also empirical justification for isolating from the secondary root not only the $\nmid f$-carrying element but also its moving subparts, i.e. disallow subextraction into the secondary root. The outcome of such an extraction is ungrammatical. This is another way in which the asymmetry between selector and selectee in a *3rd-merge* dependency manifests itself.

(4) a. *Ville has [of his daughter, I think it was a painting *t*]
b. *Ørjan has [of which daughter, do I think it was a painting *t*].

I turn now to cases where there is additional structure on top of the pivot, but the pivot NP itself has a movement feature. I want to exclude movement of the pivot of amalgams. Movement of the pivot on its own is ungrammatical and would, metaphorically speaking, lead to the amalgam being a disconnected piece of structure; pied-piping of the whole amalgam also appears to be quite degraded:

(5) a. *Chicago, Peter went to [I think it was *t*].
b. ?*[I think it was Chicago], Peter went to *t*.

Instead, I want to reserve such cases for a different phenomenon. I propose the following empirical split. With additional roots, there are two possibilities: either that root remains free, which corresponds to amalgams, or that root is reintroduced into the matrix root again. I propose that this option

[4] For this reason, the idea, as suggested by a reviewer, to let *adore* select the cleft-structure and then select for *cats* via a step of covert movement leads to wrong predictions.

only occurs when the pivot connecting both parts of the resulting cyclic graph carries a movement feature. This movement of the connecting element leads - at least from a derived tree perspective - to the breakup of the cycle. In a 1D-string, such a movement gives rise to what appears to be two distinct gaps. The phenomenon that this corresponds to is parasitic gaps (such a multidominance account of parasitic gaps has been proposed in Kasai 2010).

3.2 Parasitic gaps

3merge-4 is the rule that governs the behaviour of moving pivots[5]. Its effect is that of selection for the moving pivot without erasing the category feature of the root that hosts it - in contrast to amalgams (this can be seen in Figure 3 where *file* selects *which article* out of the adjunct). That root, ς, can either be *c* or *n/d* since parasitic gaps can occur not only in clausal elements like relative clauses or adjuncts but also in NPs as in subject parasitic gaps. A small difference to amalgams is that the host phrase of the pivot can carry a movement feature. I allow this possibility for potential movement of adjuncts or subject movement. Other than that, there are no unchecked features besides ω.

There are a number of side issues that this rule addresses as well. Upon *3rd-merging* the pivot, sub-movers $\iota_1, ..., \iota_\ell$ are 'released' so that they become accessible parts of the chain in the outcome. This appears to be empirically justified since e.g. complements of nouns that are pivots in parasitic gap constructions can be scrambled to a position lower than the pivot (though, again, only into the matrix root, not the secondary one):

(6) ?[Welche Bücher t_1]$_2$ hat [über Potsdam]$_1$
which books has about potsdam
jeder gekauft t_2 ohne je zu lesen pg_2?
everyone bought without ever to read
Which books about Potsdam did everyone buy without ever reading?

I also disallow movement of the pivot itself within the secondary root. Such a step appears unnecessary for Horn amalgams where the pivot occurs in base position within the amalgam. This also fits well with the observation that parasitic gaps in subject position are usually ungrammatical (see e.g. Mayo 1997). Under standard assumptions, subjects move to receive case. The impossibility of moving in the secondary root would exclude this for independent reasons. This ties in with another property of the pivot that I have assumed throughout the definitions in which they appeared: they always occur in complement position in the secondary root. For the *it*-cleft(-like) constructions in amalgams, this appears to be correct, as well as for parasitic gap hosts. As mentioned, subject gaps are excluded. Indirect object gaps appear to be degraded as well:

(7) ?*Which person did you send out after giving pg_1 an article?

Until a convincing need for non-complement pivots arises, I restrict *3merge-4* to complement position.

I have allowed for the presence of ω in the selector in *3merge-4* which in principle allows for amalgams in amalgams or parasitic gaps in parasitic gap hosts. Whether this is empirically justified is beyond the scope of this paper.

Let us return to the core issues. The expression that is introduced into the chain of the selector ($[t : \varsigma\gamma, u : \varepsilon]$) as a result of *3merge-4* differs from ω in that it does not contain a $\nmid f$-feature. There are three rules that govern the behaviour of such a bracketed expression. Either the pivot can move to a non-final landing site (*move3*), 'pied-piping' the whole expression with it. This would be the case if A-movement precedes $\bar{A}$-movement of the pivot.

The other two rules (*chain-merge1/2*) govern the reintroduction of the secondary root into the main root. This amounts to 'chain-internal' merge, akin to move rules, with the difference being that the expression carries an unchecked *category* feature, not a movement feature. *chain-merge1/2* describe the point in the derivation where the parasitic gap host (e.g. an adjunct like *[without reading:c, which book:-wh]*) is merged into its position in the matrix root (an application for *chain-merge1* occurs in Figure 3 where an empty vP-adjunction head ϵ selects for the adjunct). As a result, the parasitic gap host is either linearized in its final position (*chain-merge1*) or becomes a moving chain (*chain-merge2*); in both cases, the moving element that corresponds to both real and parasitic gap (*u:*ε) is released and becomes part of the chains. From there it moves to a position higher than the reintroduction site of its host, deriving the anti-c-command property of parasitic gaps. The last steps of the derivation of (2) can also be found in Figure 3[6].

[5]The rule as it stands is an abbreviation for the specifier and complement merge cases - in the latter case, $\alpha_1, ..., \alpha_k$ and ω is missing in the selector.

[6]I abstract away both from how adjunction works (treating it as normal merge) and the rightward dislocation of the

3.3 Further issues and applications

So far I have assumed that only nominals carry a $\nmid f$-feature. It is considered a core property of parasitic gaps that only NPs can correspond to them (see e.g. Culicover 2001). NPs are also prototypical pivots in amalgams; predicative adjectives e.g. are degraded as pivots in Horn amalgams (Kluck 2011, 74):

(8) ?*Bea is [I think it's blond].

Note though that Engdahl 1983 cites AP parasitic gaps as acceptable in Swedish and that amalgams in NPs on top of attributive adjectives are acceptable in some contexts ('*an [I think you can call it simple solution]*', see Kluck 2011). Further research is necessary to determine whether the restriction of $\nmid f$ to nominals is correct or needs to be relaxed at least for adjectives.

In the more restrictive system, there is only a single category that can carry the additional negative feature, i.e. either $n.\nmid n$ or $d.\nmid d$, depending on one's stance in the DP/NP-debate and/or whether one describes a DP- or NP-language (see Bošković 2008 for such a split). With the new operation, however, it is possible to propose a new solution to the structure of the DP: one assumes that NPs universally have the feature sequence $n.\nmid n$ and that verbs select for $\nmid n$, even in DP-languages, explaining why verbs can 'long-distance' select for types of NPs even in those languages (an argument against DP-structure by Bruening et al. 2018). DP-languages would differ from the system presented so far in that every DP is a '*mini-amalgam*': they require *d* to select NPs as in a *merge4* application, and it is only the resulting DP that can be the argument in a *3merge-1* rule.

$$\frac{s :: {=}n.d\gamma \qquad t \cdot n.\nmid n, \alpha_1, ..., \alpha_k}{[st : d.\nmid n.\gamma, \alpha_1, ..., \alpha_k]}\ \text{DP-merge}$$

$$\frac{s :: \#n\delta \qquad [t \cdot d.\nmid n, \alpha_1, ..., \alpha_k]}{st : \delta, \alpha_1, ..., \alpha_k}\ \text{3merge-1}^{DP}$$

The first *d* selecting $n.\nmid n$ thus has a special status, and it is only further merge with that DP that leads to amalgams proper or parasitic gap hosts.

The only purpose of *3rd-merge*, then, is to connect the two major spines, the nominal and the verbal/clausal one. In such a system, the fact that the additional root can grow further and either remain independent (amalgams) or get reconnected (parasitic gaps) is a simple consequence of the way clausal and nominal spine are merged. The three apparently distinct phenomena share a common core, and the fact that parasitic gaps and amalgams are restricted to nominals falls out as a consequence of the assumption that *3rd-merge* connects verbs and nominals and does not need to be stipulated separately.

Showing the full rule set for this system is beyond the scope of this paper, however. There are a number of empirical and theoretical issues that need to be considered. Possibilities like NP extraction out of DP as e.g. in German complicate the rules. One also needs to ensure that it is the first D selecting an NP that is turned into a bracketed ω, not a higher one. This is easier if one assumes that all additional material in NPs like adjectives and numerals are adjoined by category preserving operations and D always selects something of type $n.\nmid n$. Not all approaches assume this and would need to be dealt with differently. I therefore leave a full exploration of a system that unifies DP/NPs, amalgams and parasitic gaps for future research.

As a last point, there is also the issue that when growing amalgams or parasitic gap hosts (in a *merge4* step), the verb needs to select via $=n$ or $=d$. Thus one would need to allow optionality in the way verbs select arguments ($\nmid n/{=}d$) which appears to unnecessarily bloat their lexicon entry. However, this is independently necessary if one assumes that (weak) pronouns are just a single head *d* [7]. A stronger but related argument for the variable nature of selection comes from a number of verbs that disallow weak pronouns, Postal's 1994 so-called anti-pronominal contexts (9-a). Strikingly, it is exactly those verbs that cannot occur in parasitic gap hosts (9-c) even though they do allow wh-extraction (9-b). Both apparently unrelated facts are derived together by the assumption that the lexicon entries of this class of verbs lack the $=d/{=}n$ option:

(9) a. *She likes the colour black, so she painted the door it.
b. What colour$_1$ did she paint the door t_1?
c. *What colour$_1$ did she grow to hate t_1 after painting her door pg_1?

This is also yet another example of an asymmetry

adjunct in this example.

[7] Verbs still cannot select via $=d$ for 'full' DPs with lexical content in matrix roots since they would then contain unchecked $\nmid n$, preventing the derivation to converge.

between the roots since the mode of selection for the pivot appears to differ. An in-depth investigation of the empirical facts concerning this asymmetry is beyond the scope of this paper, however.

4 Discussion

To summarize, I introduced a new operation, *3rd-merge*, to Minimalist Grammars. By postulate, only nominals can carry $\nmid f$ so the new long-distance dependency holds between nominal arguments and their selectors. The main effect of the new operation is to allow selection of an item from within an additional root without establishing a syntactic relation to any other part of that root. Additional roots can either remain independent, which corresponds to the phenomenon of Horn amalgams, or they can be reintroduced into the matrix root, which results in parasitic gap constructions after movement of the pivot in the resulting cyclic structure.

I want to discuss a number of commonalities and differences between *3rd-merge* and other extensions of minimalist grammars. What *3rd-merge* and sidewards movement (Nunes 1995, especially in the formalization by Stabler 2006) have in common is the general idea of further relaxing resource sensitivity. In the sidewards movement system in Stabler (2006), however, a category feature e.g. can be re-used a potentially unbounded number of times. The system set up here does not give up resource sensitivity completely but only allows one further type of re-use of an expression, besides being merged and moved, thereby stipulating a third dependency type. The third type of re-use leads to the growth of an additional root which is distinct from the possibilities of sidewards movement. Just as movement is not 'just' a reuse of category features but a dependency (related to but nonetheless) distinct from merge with its own properties and restrictions, it is important in my opinion to equally separate this third reuse of expressions. This way one can investigate the properties and restrictions of this new dependency in their own right.

If resource-sensitivity needs to be relaxed further, e.g. for multiple parasitic gap constructions, one has more control over which features exactly are to be changed in that way. Whether it is *merge*, *move* or *3rd-merge* features that can be reused might have different empirical consequences.

Torr and Stabler (2016), building on Kobele (2008), extend MGs to deal with ATB-movement (among other things). The idea behind these approaches is the unification of the identical but distinct movers of both conjuncts. Those approaches are then extended to cover other one-to-many dependencies such as control and parasitic gaps. Parasitic gaps, however, are markedly different from ATB-constructions as demonstrated in Postal (1993). They are not confined to coordinations but are subject to a number of restrictions irrelevant for ATB, such as a restriction to $\bar{A}$-movement, a categorial restriction to NPs, the anti-pronominal condition shown in (9) and many others. Parasitic gaps are optional and their position can be filled, contrary to ATB-patterns where all gaps are obligatory and mutually depend on each other, i.e. a mutual symbiosis compared to an asymmetric parasitism. There is also the asymmetry in subextraction (see (6)) that is non-trivial in a system that treats the origin of unified movers on equal footing. For these reasons I treat the asymmetries of parasitic gaps as a different phenomenon, not a subtype of ATB-movement.

The properties of amalgams are another central reason to adopt a system as presented here. There is convincing evidence that amalgams contain an undominated, independent secondary root (Kluck 2011, ch.3), a structure the above approaches cannot currently generate. In the present approach, a head can select for an element from within a different root without, however, connecting with the rest of the root. Amalgams also exhibit restrictions and asymmetries that are shared by parasitic gaps, such as a putative categorial restriction to NPs and (sub)extraction asymmetries (see (4)). Since these phenomena pattern together and they can both be derived by a system that allows multiple roots, it is useful to derive them with the same mechanism while treating the more symmetrical ATB-phenomenon as distinct.

What distinguishes *3rd-merge* from parallel merge (Citko 2005, Citko 2011), apart from a formal implementation, is that it is not 'parallel' or symmetric. In parallel merge, a head A and a head C that merge with phrase B both stand on equal footing. *3rd-merge* introduces an asymmetry between selector and selectee. This property is shared by grafts (van Riemsdijk 2006, van Riemsdijk 2010), the operation that is its closest match. van Riemsdijk uses this operation mainly to derive properties of free relatives and transparent free relatives ('*She ate what she called egg fried rice.*') but also Horn amalgams. van Riemsdijk notes empir-

ical asymmetries but remains agnostic as to how they come about (see e.g. van Riemsdijk 2006, fn.8, 'all trees are equal'). For *3rd-merge*, the asymmetry is built into the definition of the operation: the selector is always part of the matrix root while additional structure on top of the NP is always a secondary root that is integrated into the main structure. Further asymmetries are part of the definition, such as the impossibility of extraction and subextraction of the pivot into the secondary root. Graft can apply at any stage but must do so long before the whole clause is built for phase considerations (van Riemsdijk 2006, ch. 4.3). I proposed the opposite for *3rd-merge* since merge of an intermediate stage of a secondary root would lead to a proliferation of unchecked features that are difficult to track in the algebraic definition presented here.

Before closing this article, I want to mention two further issues that need to be addressed. One is the linearization of amalgams. Not only were the pivots considered so far the most deeply embedded complement, they were also the most rightward element in the string. This would be different in SOV amalgams or with extraposed adverbials:

(10) ?Peter hat [ich glaub es war die Katze gewesen]
Peter has I think it was the cat been
gestreichelt.
petted.
Peter petted I think it was the cat.

This cannot be derived in the system set up so far. One reason for this is that the algebraic definition used here does two things at once: regulate the feature calculus *and* linearize the string. A more fine-grained approach should be able to treat those matters separately.

The last question concerns the expressive power of the grammar presented so far. Though I assume it to be the case, showing the equivalence to MCFGs would be reassuring. Apart from empirical considerations, it might be relevant for that purpose to determine whether to allow ω in *3merge-2'/4*, i.e. whether it is safe to allow unbounded nested amalgams/parasitic gaps. The same goes for the question whether there should be an SMC equivalent for the structure $[t : \varsigma\gamma, u : \varepsilon]$. Occurrence of more than one such element in an expression might lead to unwanted indeterminacies. As a last point, it would be of interest to know whether MGs with *3rd-merge* but without (remnant) movement allow generation of non-context free patterns.

References

Željko Bošković. 2008. What will you have, DP or NP? In *Proceedings of NELS 37*, volume 1, pages 101–114.

Benjamin Bruening, Xuyen Dinh, and Lan Kim. 2018. Selection, idioms, and the structure of nominal phrases with and without classifiers. *Glossa: a journal of general linguistics*, 3(1).

Barbara Citko. 2005. On the Nature of Merge: External Merge, Internal Merge, and Parallel Merge. *Linguistic Inquiry*, 36(4):475–496.

Barbara Citko. 2011. Multidominance. In Cedric Boeckx, editor, *The Oxford Handbook of Linguistic Minimalism*, chapter 6, pages 119–142. Oxford University Press.

Peter W. Culicover. 2001. Parasitic Gaps: A History. In *Parasitic Gaps*, pages 3–68. Oxford University Press.

Elisabet Engdahl. 1983. Parasitic gaps. *Linguistics and Philosophy*, 6(1):5–34.

Hironobu Kasai. 2010. Parasitic Gaps Under Multiple Dominance. *English Linguistics*, 27(2):235–269.

Marina Elisabeth Kluck. 2011. *Sentence amalgamation*. Ph.D. thesis, University of Groningen.

Gregory M. Kobele. 2008. Across-the-board extraction in Minimalist Grammars. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+9)*, pages 113–120, Tübingen, Germany. Association for Computational Linguistics.

George Lakoff. 1974. Syntactic Amalgams. In *Papers from the Tenth Regional Meeting of the Chicago Linguistic Society*, volume 10, pages 321–344. Chicago Linguistic Society.

Pilar García Mayo. 1997. Non-Occurrence of Subject and Adjunct Parasitic Gaps. *Atlantis*, 19(2):125–133.

Jairo Nunes. 1995. *The Copy Theory of Movement and Linearization of Chains in the Minimalist Program*. Ph.D. thesis, University of Maryland.

Paul M. Postal. 1993. Parasitic gaps and the across-the-board phenomenon. *Linguistic Inquiry*, 24(4):735–754.

Paul M. Postal. 1994. Parasitic and pseudoparasitic gaps. *Linguistic Inquiry*, 25(1):63–117.

Edward Stabler. 1997. Derivational minimalism. In *Logical Aspects of Computational Linguistics*, pages 68–95. Springer Berlin Heidelberg.

Edward P. Stabler. 2006. Sidewards without copying. In *Proceedings of the 11th conference on Formal Grammar*, pages 157–170.

$$\dfrac{\dfrac{\text{was} :: {=}n.{=}n.v \qquad \text{cats} : n.{\not=}n}{\text{was} : {=}n.v, [\text{cats} : {\not=}n]}\,\text{merge4} \qquad it :: n}{\text{it was} : v, [\text{cats} : {\not=}n]}\,\text{merge2}$$

$$\vdots$$

$$\dfrac{\text{adore} : \#n.\#n.v \qquad \text{I think it was} : c, [\text{cats} : {\not=}n]}{\text{adore I think it was cats} : \#n.v}\,\text{3merge-1'}$$

$$\vdots$$

$$\text{Joscha adores I think it was cats} : c$$

(a) partial derivation of an amalgam as in (1)

$$\dfrac{\text{reading} :: {=}n.v \qquad \text{which article} : n.{\not=}n.{-}wh}{\text{reading} : v, [\text{which article} : {\not=}n.{-}wh]}\,\text{merge4}$$

$$\vdots$$

$$\dfrac{\text{file} :: \#n.\#n.v \qquad \text{without reading} : c, [\text{which article} : {\not=}n.{-}wh]}{\text{file} : \#n.v, [\text{without reading} : c, \text{which article} : {-}wh]}\,\text{3merge-4}$$

$$\vdots$$

$$\dfrac{\epsilon \text{ the manager file} := c.v, [\text{without reading} : c, \text{which article} : {-}wh]}{\text{the manager file without reading} : v, \text{which article} : {-}wh}\,\text{chain-merge1}$$

$$\vdots$$

$$\dfrac{\text{did the manager file without reading} : {+}wh.c, \text{which article} : {-}wh}{\text{Which article did the manager file without reading} : c}\,\text{move1}$$

(b) partial derivation of a parasitic gap as in (2)

Figure 3: Example derivations

Edward P. Stabler and Edward L. Keenan. 2003. Structural similarity within and among languages. *Theoretical Computer Science*, 293(2):345–363.

John Torr and Edward P. 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.

Henk van Riemsdijk. 2006. Grafts follow from merge. In *Phases of Interpretation*, pages 17–44. Mouton de Gruyter.

Henk van Riemsdijk. 2010. Grappling with graft. In *Structure Preserved*, Linguistik Aktuell, pages 289–298. John Benjamins Publishing Company.

A TSL Analysis of Japanese Case

Kenneth Hanson
Department of Linguistics
Stony Brook University
Stony Brook, NY 11794, USA
kenneth.hanson@stonybrook.edu

Abstract

Recent work in subregular syntax has revealed deep parallels among syntactic phenomena, many of which fall under the computational class TSL (Graf, 2018, 2022). Vu et al. (2019) argue that case dependencies are yet another member of this class. But their analysis focuses mainly on English, which is famously case-poor. In this paper I present a TSL analysis of Japanese, which features a much wider range of case-marking patterns, adding support to the claim that case dependencies, and by extension syntactic dependencies, are TSL.

1 Introduction

Work on the computational complexity of strings has identified a rich hierarchy of subregular classes and shown that phonological patterns are among the simplest possible (Heinz, 2018). Local dependencies fall under the class of *strictly local* (SL) languages while most long-distance dependencies fall within a superclass of SL known as *tier-based strictly local* (TSL). These findings are of interest to both computational and general linguistics as they make strong typological predictions and inform development of learning algorithms (Lambert et al., 2021). A tantalizing possibility is that the tree-based equivalents of the string classes might reveal the same result in syntax. Graf (2018) generalizes SL and TSL to trees, and subsequent work (Graf and Shafiei, 2019; Vu et al., 2019; Graf, 2022, a.o.) presents preliminary evidence that many disparate syntactic phenomena are indeed TSL. But confirming this hypothesis requires much additional work, because the abstractness of syntactic representations makes it difficult to claim with certainty what structures are possible.

This paper focuses on the syntactic distribution of morphological case, which I define to be those heads or features realized by case morphology. In other words, we are not interested in the raw surface forms (which may exhibit accidental syncretism), but in the systematic distinctions among nominals made on the basis on their syntactic context. Vu et al. (2019) provides a proof of concept for a TSL analysis of case, focusing primarily on English. This work provides an analysis of Japanese, which features a much richer range of case patterns, including: (1) case marking conditioned by temporal properties of verbs, (2) lexical and structural dative case, (3) long-distance case marking in embedded clauses, and (4) case alternations in complex predicates. In addition to strengthening the claim that the syntactic distribution of case is TSL, the analysis also shows that case patterns that might otherwise be considered complex or surprising are in fact quite simple from a computational perspective.

The remainder of this paper is structured as follows. Section 2 introduces the computational background for establishing the TSL nature of syntactic dependencies. Section 3 provides an overview of the basic case patterns in Japanese, and proposes a set of descriptive generalizations. In Section 4, I define TSL grammars which encode these generalizations, then show how the analysis extends easily to more complex structures. Section 5 concludes.

2 Computational background

2.1 SL and TSL string languages

A strictly local (SL) language is characterized by a set of *forbidden substrings* of a fixed length k. For example, an SL grammar enforcing strict CV syllable structure consists of the set {V, CC, VV, C}, where $ stands for beginning/end of string. Words in this language include CV and CVCV but not CVCCV (which contains CC) or CVC (C$). Each forbidden substring is of length 2, making this a *strictly 2-local* (SL-2) language.[1]

TSL is a generalization of SL in which certain

[1] An equivalent definition of SL utilizes sets of *permissible substrings* of fixed length k, {C, CV, VC, V} in the case of the present example. Under this definition, a word is well-formed iff all of its length k substrings are well-formed.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 15-24.
Amherst, Massachusetts, June 15-17, 2023

symbols are ignored. The remaining symbols are projected onto a *tier* in which elements that were formerly separated become adjacent, allowing a restricted type of long-distance dependency: a string is well-formed iff its tier conforms to a given SL grammar. A simple example from phonology is (symmetric) consonant harmony. Assuming an alphabet {a, m, s, ʃ}, we project only {s, ʃ}, and ban the substrings {sʃ, ʃs}. Words like 'samaas' (tier: 'ss') and 'ʃaʃaʃa' (ʃʃʃ) are part of this language but 'ʃamas' (ʃs) and 'saʃaʃa' (sʃʃ) are not. Since the forbidden substrings on the tier are of length 2, this language is TSL-2.

2.2 TSL in syntax

Graf (2018) generalizes TSL from strings to trees as follows. First, we project a *tree tier* which retains a subset of the original nodes, preserving dominance and precedence relations. The daughters of each node on the tier are then regulated by a TSL string language. This means that there are two opportunities to project a tier; we will take advantage this in of our treatment of adjuncts in Section 4.8.

Somewhat more formally, a TSL tree language is defined in terms of two functions: a *tier projection function* specifying which nodes to retain on the tree tier, and a *daughter string language function* which determines the constraints on the daughter string of each node.[2] Each function considers a finite local context of the argument node. I will use only the label of the node itself—a context of size 1, with one exception as discussed in Section 4.7.

For the syntactic formalism, I follow recent work (Graf, 2022; Graf and Shafiei, 2019) by adopting Minimalist Grammar (MG, Stabler 1997) dependency trees. These trees record the order of Merge steps in a Minimalist derivation: the rightmost child of a head is its complement, and other children are specifiers. Dependency trees are more compact than other representations while containing all necessary information about the derivation.

Consider the Japanese example in (1). This is a simple transitive sentence, in which the subject *Taroo* is followed by the nominative case marker *ga* and the object *piano* is followed by the accusative marker *o*. An X′-style phrase structure tree for this sentence is shown on the left of Figure 1. Details of the syntactic analysis will be introduced in Section 4. For now, it suffices to note that the case marker is the head of a K(ase) phrase, and that the subject

[2]See Graf and Kostyszyn (2021) for a full definition.

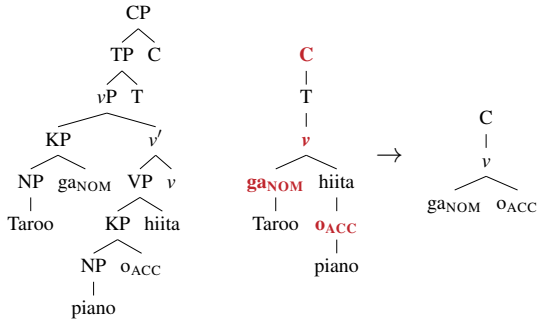


Figure 1: Left: phrase structure tree. Right: dependency tree and tier projection enforcing accusative constraint. Nodes of category K, C, and *v* are projected. It is required that every *o* has a *ga* among its left sisters.

KP asymmetrically c-commands the object KP.

(1) Taroo ga/*o piano o/*ga hiita.
Taroo NOM/ACC piano ACC/NOM played
'Taroo played the piano.'

On the right of Figure 1 is the dependency tree corresponding to the phrase structure tree, along with the case tier projection. Each node in the dependency tree is a lexical item, taking the place of a head and all of its projections in the phrase structure tree. *v* has two daughters, corresponding to its specifier (the subject, headed by the case marker) and its complement (VP, headed by the verb). Other nodes have only a single child, corresponding to their complements. A full dependency tree would display the features of each node; for brevity I omit everything but the node label and relevant features such as case, using the category as the label of empty elements such as C/T/*v*.

This brings us to the tier projection. The general approach taken in this paper will be to construct a tier such that all nominals in some case licensing domain become daughters of the domain node, and to state the constraints on case configurations over the daughter strings of the domain nodes. In the present example, the relevant constraint (simplified) is that the accusative marker *o* must be c-commanded by nominative *ga* in the same clause. To enforce this constraint, we project a tier which includes all nodes of category *v*, K, and C. Since dominance and precedence are preserved, *ga* and *o* become daughters of *v*. We then require that *ga* be a left sister of *o*.[3] The TSL grammar for the daughter string language of *v* will thus ban substrings such as *o ga*. The full analysis, which contains

[3]In principle it is possible for a left sister on the tier not to be c-commander in the dependency tree. In practice this turns out not to be an issue. See Section 4.8 for an example.

many additional constraints over several tiers, will be fleshed out in Section 4.

It is worth noting that in an MG dependency tree all elements appear in the position of first merge—when a nominal has moved, such as by passivization or scrambling, only its base position is considered for purposes of case licensing. This assumption has been adequate for all phenomena examined in this framework to date, and it also seems to be appropriate for case in Japanese. Scrambling, for example, is widely understood to preserve case marking. Even when a certain case correlates with movement (as in some analyses of differential object marking), it is usually possible to predict the case of nominal based on its context and its other features (e.g. definiteness). Since this issue does not arise in the Japanese data, I say no more here.

3 Basic case patterns

Japanese has four core cases, marked by the suffixes *ga* (nominative), *o* (accusative), *ni* (dative), and *no* (genitive). Their prototypical functions are similar to German and other Indo-European languages: subjects receive nominative case, direct objects receive accusative, and indirect objects receive dative, while complements and possessors of nouns are genitive. Examples of simple intransitive (2a), transitive (2b), and ditransitive sentences (2c) are given below, along with examples of a nominal complement (2d) and possessor (2e).[4] All examples are presented in topic-less sentences since topic marking masks the underlying case.[5]

(2) a. Taroo ga hasitta.
Taroo NOM ran
'Taroo ran.'

b. Taroo ga piano o hiita. (=1)
Taroo NOM piano ACC played
'Taroo played the piano.'

c. Jin ga Yumi ni hon o ageta.
Jin NOM Yumi DAT book ACC gave
'Jin gave Yumi a book.'

d. Taroo ga [yama no e] o mita.
Taroo NOM mountain GEN picture ACC saw
'Taroo saw a picture of a mountain.'

e. Taroo no hon
Taroo GEN book
'Taroo's book'

[4]Data is adapted from (Miyagawa, 1989) unless noted otherwise.

[5]Abbreviations: NOM = nominative, ACC = accusative, DAT = dative, LD = lexical dative, GEN = genitive, APPL = applicative, NPST = non-past, PASS = passive, IPASS = indirect passive, CAUS = causative.

In general, nominative, accusative, and dative case are available for arguments of verbs, and the number of arguments predicts what their cases should be: if there is one argument then it is nominative; if there are two then the latter is accusative, and if there are more than two then the middle nominals are dative. This is also true for complex verbal predicates, with some complications (discussed in Sections 4.6 and 4.7). Conversely, arguments of nouns are usually genitive no matter how many there are. An example of a noun phrase multiple genitive arguments is given in (3) below.

(3) Taroo no yama no e
Taroo GEN mountain GEN picture
'Taroo's picture of a mountain'

While these are the canonical patterns, several others are possible. Some transitive verbs take a dative object rather than the usual accusative (4a). Additionally, stative verbs such as *dekiru* 'can do' take a nominative object, and allow dative and/or nominative for the subject (this varies depending on the exact verb), yielding dative-nominative (4b) and double nominative (4c) structures. Transitive adjectives and complex verbs formed with a stative suffix also allow nominative objects.

(4) a. Taroo ga Yumi ni atta.
Taroo NOM Yumi DAT met
'Taroo met Yumi.'

b. Yumi ni tenisu ga dekiru.
Yumi DAT tennis NOM can.do
'Yumi can play tennis.'

c. Yumi ga tenisu ga dekiru.
Yumi NOM tennis NOM can.do
'Yumi can play tennis.'

Of the four cases, nominative has the widest distribution. As we will see later (Sections 4.5 and 4.7), it can also be replaced with another case when a verb or adjective and its arguments are embedded in a larger structure. Thus, it makes sense to treat nominative as the *default* case, appearing when no other condition applies.

To briefly summarize, the case that marks a nominal in some domain depends primarily on (1) the category of the domain and (2) the position of that nominal relative to others in the domain. Additionally, certain predicates specify that one of their arguments must be dative rather than the case that would otherwise be expected. Specifically, we could say that accusative and genitive are *structural* cases (i.e. licensed by the structural context); some instances of dative are structural while oth-

ers are *lexical* (licensed by specific lexical items); finally, nominative is the *default*.

I am not aware of any work in the syntactic literature that analyzes the entire case system of Japanese in this manner. However, the individual patterns are well-known, and the analysis is a direct application of ideas from dependent case theory (Marantz, 2000; Baker and Vinokurova, 2010). The primary purpose of this paper is to show that the generalizations outlined above are easily implemented using a TSL grammar, and that they can be extended to more complex constructions with little or no modification. As a computational analysis, it essentially descriptive in nature, and in principle compatible with a variety of theories of case licensing. At the same time, most of the patterns discussed here (or close analogues) can also be found in other rich case-marking languages, so there is good reason to believe that the approach should generalize beyond Japanese.

4 Analysis

4.1 Preliminaries

In this section, I will formalize the generalizations made in the previous section. To begin, I lay out a few syntactic assumptions. First, clauses are assumed to have the following functional hierarchy:

$$\text{C} > \text{T} > (\textsc{pass}) > (\textsc{caus}) > v > (\textsc{appl}) > \text{V}$$

In essence, this is a modern version of the "biclausal" analysis for the passive and causative constructions. These heads may be considered subtypes of *v*, labeled separately for convenience. Next, goals of ditransitive verbs may appear in two positions: low goals are PP daughters of VP, while high goals are KP daughters of an applicative head (Miyagawa and Tsujioka, 2004). This fact will be relevant to the analysis of passivization in Section 4.6. Finally, nominals are treated as NPs, with case markers occupying a higher KP.[6]

The remainder of this section is structured as follows. First, I introduce three tree tiers corresponding to structural cases licensed in the verbal domain, the nominal domain, and lexical case. Next, I consider more complex constructions, including embedded clauses, passives, and causatives, making several small revisions. From there, I refine the analysis to handle adjuncts, and address a potential problem involving coordination.

[6]PPs take an NP complement instead of a KP. PPs may alternatively be analyzed as KPs bearing semantic case. Such cases do not need to be licensed syntactically.

4.2 Accusative and structural dative case

First, we define a tier to license structural cases in the verbal domain: accusative and dative. On this tier we project non-stative *v* and all K heads. We also project C heads in order to limit the case licensing domain to a single clause; while *v* in the embedded clause will normally do this, it will not always be present on the tier.

The constraints on the tier are, roughly: (1) the rightmost of two or more K children of *v* must bear accusative, (2) the middle of three or more K heads must bear dative, and (3) no other K heads may bear accusative or dative. Other K heads are underspecified; if not specified as genitive or lexical dative on the relevant tiers they become nominative by default. This includes all subjects as well as objects of stative verbs (since stative *v* is not projected).

An example for the simple transitive sentence in (1) is given in Figure 2a. Since non-stative *v* is projected, the tier is unchanged from the example in Section 2.2. The daughter string of *v* satisfies the constraints just mentioned: the accusative K is the rightmost of two K children of *v* and the nominative K meets the elsewhere criterion. Further examples of well/ill-formed tiers are given in Figures 2b and 2c, respectively.

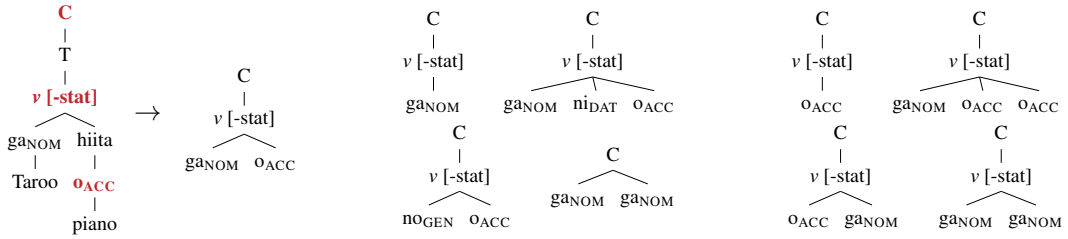
We must also take into account the fact that lexical datives are allowed as direct objects. It turns out that many verbs are compatible with either an accusative or dative object, with a difference in temporal properties (cf. Fukuda, 2007); I assume such verbs to be optional licensors of lexical dative case. The full tier definition is given below.[7]

(5) **Verbal case tier (initial version)**
Project categories: {*v*[-stat], K, C}
Daughter string languages:
v: $\{\textsc{nom}, \textsc{gen}, \textsc{ld}\} \cdot (\textsc{dat}^* \cdot \{\textsc{acc}, \textsc{ld}\})$
K/C: $\overline{\{\textsc{acc}, \textsc{dat}\}}^*$

For clarity, all daughter string languages are defined using regular expressions. Since it may not be immediately obviously that these languages are TSL (or SL), I also provide grammars for the verbal tier:

- The daughter string language of *v* is SL-2. The grammar (set of forbidden substrings) is {$ DAT, $ ACC, NOM NOM, NOM GEN, GEN

[7]String languages are notated using regular expressions. NOM/ACC/etc. stand for a K head bearing said case. A dot (·) represents concatenation. Set braces represent a choice among alternatives. An overbar represents set complement.



(a) Dependency tree and verbal case tier for a transitive sentence. The subject licenses the accusative object to its right.

(b) Other licit verbal case tiers. From left to right: intransitive verb, ditransitive verb, genitive subject, stative verb.

(c) Examples of illicit verbal case tiers. In order: lone accusative, double accusative, accusative before nominative, non-stative double nominative.

Figure 2: Examples of licit and illicit verbal case tiers.

NOM, GEN GEN, DAT $, DAT NOM, DAT GEN, ACC NOM, ACC GEN, ACC DAT, ACC ACC, ACC LD, LD NOM, LD GEN, LD LD}.

- The daughter string language of K/C is SL-1. The grammar for this language is {ACC, DAT}.

Small modifications to the verbal case tier will be required; the revised tier definition is given in Section 4.7. Also, while the current daughter string languages are SL, they will later be converted to TSL to accommodate adjuncts (Section 4.8).

4.3 Genitives

Next, we turn to genitives, which have the simplest distribution: as a first approximation, all KPs in the domain of a nominal are genitive, and no others. We construct the genitive tier as follows:

(6) **Genitive case tier (initial version)**
Project categories: {N, K, C}
Daughter string languages:
N: $\{\text{GEN}, \text{N}, \text{C}\}^*$
K/C: $\overline{\{\text{GEN}\}}^*$

The tier projection for (2d), in which the object nominal contains a genitive complement, is shown in Figure 3a. There is only a single K child the noun *e* 'picture', and it bears GEN as required, so the tier is well-formed. There are no restrictions on the other KPs, though they could of course be ruled out on other tiers.

Subjects of certain embedded clauses appear in genitive case, an apparent exception to the current tier definition; this will require modification as discussed in Section 4.5. Another issue worth noting is that the particle *no* can appear between PPs and their head nouns, as in example (7) below. This *no* is traditionally considered to be a marker of adnominal modification rather than a case particle. Fortunately, we can abstract away from this issue. If these instances of *no* are case particles, then they still adhere to the constraint as stated; if not, then the constraint simply does not apply.

(7) otera e no michi
temple to NO road
'the road to the temple'

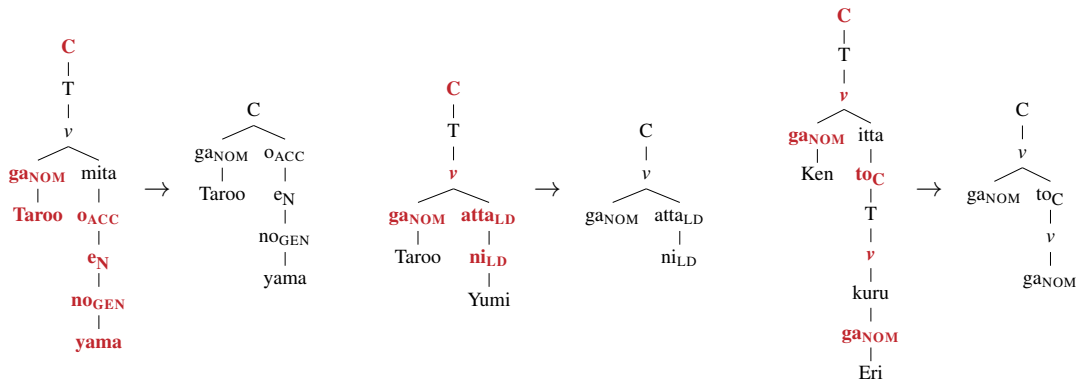
4.4 Lexical datives

The third case tier controls the distribution of lexically dative-marked nominals. While we could reasonably leave lexical case to be handled by the selection (i.e. subcategorization) mechanism, it is worth demonstrating that both structural and lexical case can be regulated in a unified manner if desired. Lexical dative may be assigned to an argument of either *v* or V depending on the verb, but only *v* appears on the verbal case tier, so a new tier is required which projects both. On this tier, verbal heads licensing a lexical dative KP must have exactly one such daughter; lexical dative KPs may appear nowhere else. The tier is defined as follows:

(8) **Lexical case tier**
Project categories: {*v*, V, K, C}
Daughter string languages:
v/V (LD licensor): $\overline{\{\text{LD}\}}^* \cdot \text{LD} \cdot \overline{\{\text{LD}\}}^*$
v/V (non-LD licensor): $\overline{\{\text{LD}\}}^*$
K/C: $\overline{\{\text{LD}\}}^*$

The tier projection for (4a), in which the verb requires a lexical dative object, is given in Figure 3b. The only KP child of V is a lexical dative, so the tier is well-formed. If the dative licensor was *v* then the subject would need to be dative instead.

While it might be desirable to use the same feature for both structural and lexical datives, this would prevent structural datives from being ruled out in subject and direct object position. Since lexical datives differ in behavior from other nominals (they cannot be passivized in Japanese, for example), such a distinction seems appropriate. In effect, we are treating structural and lexical dative



(a) Genitive tier for nominal complement. (b) Lexical case tier for dative object verb. (c) Verbal case tier for embedded CP.

Figure 3: Example tier projections for genitive and lexical dative, and an embedded clause.

as different cases.

One question that this analysis raises is what should happen if there are two KP children of the same lexical dative case licensor, in particular V. As far as I am aware, this situation never arises in Japanese. If it does in other languages, then the grammar must specify which KP should be dative.

Now, having defined three tiers modeling the canonical uses of the four core cases, we will consider more complex structures, and see that the system can handle them with minimal adjustment.[8]

4.5 Embedded clauses

There are several types of finite embedded clauses in Japanese. By default, these show the same case marking as matrix clauses, but under certain circumstances the embedded subject may be marked accusative or genitive.

We will first confirm that the analysis works correctly for the basic pattern. Examples of two types of finite embedded clauses are given in (9) below. Here, *to* is analyzed as a complementizer, while *koto* is a noun taking a CP complement.

[8] As noted by a reviewer, it has been suggested for phonology that when a dependency involves multiple tiers, the tier alphabets are either nested or disjoint, but never incomparable (Aksënova and Deshmukh, 2018). Since lexical dative is always assigned locally the tier projection could be expanded to all categories (in effect, an SL tree grammar), making it a superset of the others. It may also be possible to combine the verbal and genitive case tiers into a single tier, in which case the generalization would be upheld. But generally speaking we when we look at the whole system (not just case licensing) we expect overlapping tiers (Thomas Graf, p.c.).

(9) a. Ken ga [Eri ga kuru to] itta.
Ken NOM Eri NOM come C said
'Ken said that Eri will come.'

b. Eri ga [Ken ga tegami o okutta ∅]
Eri NOM Ken NOM tegami ACC sent C
koto o sitteiru.
thing ACC know
'Eri knows that Ken sent the letter.'

Since we project C on the tier, a new case domain is created for each embedded clause, resulting in the same case configuration as in a matrix clause. As an example, the tier projection for sentence (9a) is shown in Figure 3c. While projecting C may seem redundant, it is necessary because *v* is not projected on the verbal tier when it is stative. It also provides the basis for the analysis of the alternative case marking patterns, which we now turn to.

In the Japanese ECM construction, the embedded subject appears to take accusative case (10). If this nominal was a matrix object binding a *pro* subject in the embedded clause (a prolepsis analysis) then there would be nothing to explain, but Kishimoto (2018) argues that at least some ECM subjects are genuine. Similarly, in *ga-no conversion* the subject of a nominal clause takes genitive case (11). Both structures are also grammatical with a nominative embedded subject.

(10) Finite ECM (Kishimoto 2018)
Ken ga [Eri {ga/o} kawaii to] omotteiru.
Ken NOM Eri {NOM/ACC} be.cute C think
'Ken thinks that Eri is cute.'

(11) Ga-no conversion (Maki and Uchibori 2008)
Eri ga [Ken {ga/no} kita ∅] riyuu
Eri NOM Ken {NOM/GEN} came C reason
o sitteiru.
ACC know
'Eri knows the reason that Ken came.'

This variable cross-clausal case marking may

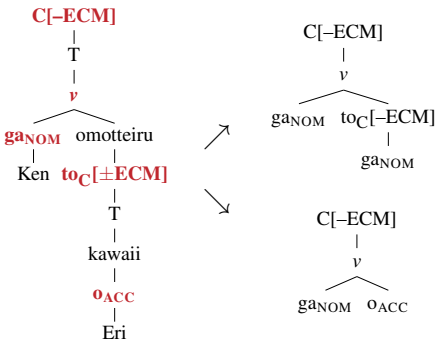


Figure 4: Verb case tier for embedded clause with and without ECM. C[+ECM] is not projected, so the embedded subject becomes part of the higher case domain.

seem mysterious, but in fact all that is needed to derive the patterns is to selectively ignore the embedded C head. The easiest way to do this is to posit that the relevant lexical predicates may select a C head with a special feature, call it [ECM]. Such C heads are not projected, similar to our treatment of stative verbs. This approach also has precedent in work which attributes variation in cross-clausal dependencies to the feature composition of the complementizer (cf. Lohninger et al., 2022). Indeed, Hiraiwa (2001) claims that ga-no conversion involves a special complementizer.

Example tier projections for (10) are shown in Figure 4 (the treatment of (11) is exactly parallel). Revised tier definitions are provided in Section 4.7. For simplicity, I treat the subject of an adjective as its complement, and ignore the aspectual morphology of *omotteiru*. All said, the facts about embedded clauses are handled quite well under the TSL perspective.

4.6 Passives

Next, we examine complex predicates within a single clause, formed with the passive suffix *-rare* and the causative suffix *-sase*. The passive suffix itself has at least two functions: the *direct passive*, which decreases the valency of a transitive verb by eliminating the agent, and the *indirect passive*, which increases valency. The literature disagrees on exactly how many distinct lexical items exist (see Ishizuka 2017 for an overview). I assume two homophonous passive suffixes corresponding to the two major functions. Recall also that I assume these suffixes to realize distinct functional heads, though the analysis could also work with verbs bearing 'passive' and 'causative' features.

The direct passive will be the focus on this section; the indirect passive will be discussed together with causatives. An example is given in (12).

(12) Active/passive transitive verb
  a. Sensei ga gakusei o hometa.
  teacher NOM student ACC praised
  'The teacher praised the student.'
  b. Gakusei ga (sensei ni) homerareta.
  student NOM teacher by praised.PASS
  'The student was praised (by the teacher).'

The object of a passivized transitive verb is promoted to the subject, and receives nominative case. These facts are straightforwardly understood under the common assumption that agent is not projected in Spec-*v*P in passives, and that the optional *by*-phrase is an adjunct PP. Miyagawa (1989) argues that this is indeed the case in Japanese.

For ditransitive verbs, there are two possibilities: either the (higher) goal is promoted, or the (lower) theme is promoted. Example (13) shows an active ditransitive verb along with the corresponding goal (13b) and theme (13c) passives (optional *by*-phrases are omitted).

(13) Active/passive ditransitive verb
  a. Mari ga kodomo ni okasi o ataeta.
  Mari NOM child DAT candy ACC gave
  'Mari gave the child candy.'
  b. Kodoma ga okasi o ataerareta.
  child NOM candy ACC gave.PASS
  'The child was given candy.'
  c. Okasi ga kodomo ni ataerareta.
  candy NOM child DAT gave.PASS
  'The candy was given to the child.'

The goal passive (13b) requires no special explanation assuming that dative case here is structural. Once the agent is eliminated, there are only two arguments, effectively creating a transitive verb. There is an elegant solution for the theme passive as well. Recall that the goal of a ditransitive verb may occupy one of two positions, and that the higher position is a KP while the lower position is a PP. Thus, it should be possible to target the direct object for promotion by selecting the low goal structure. So we see that the facts about passives fall out naturally in this analysis. Dependency trees and verbal case tiers for the ditransitive goal passive and theme passive are shown in Figure 5.

4.7 Causatives

As our final case study, we consider the causative construction. The causative morpheme *sase* is compatible with verbs of any valency. Causative equivalents of the examples in (2) are given in (14) be-

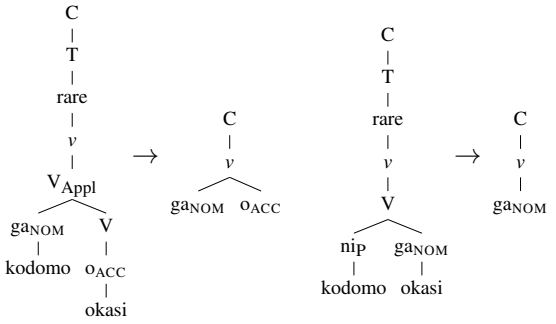


Figure 5: Verbal case tiers for passives of ditransitive. Left: goal passive (goal is a KP). Right: theme passive (goal is a PP).

low. The causee of an intransitive verb may be accusative or dative, corresponding to the *make* or *let* interpretations, respectively. For other verbs the causee must be dative, and either interpretation is possible. I set aside these semantic details.

(14) a. Ken ga Taroo {ni/o} hasiraseta.
Ken NOM Taroo DAT/ACC ran.CAUS
'Ken made/let Taroo run.'

b. Ken ga Taroo ni piano o hikaseta.
Ken NOM Taroo DAT piano ACC played.CAUS
'Ken made/let Taroo play the piano.'

c. Ken ga Jin ni Yumi ni hon o
Ken NOM Jin DAT Yumi DAT book ACC
agesaseta.
gave.CAUS
'Ken made/let Jin give Yumi a book.'

In an intransitive sentence the causee may be dative without an accompanying accusative object, suggesting lexical dative case. Additional arguments appear in the expected cases, suggesting that these are the usual structural cases. But in the present system it is the causer that would receive dative case from *sase*, not the causee. Furthermore, it is possible to passivize a causative, in which case the causee becomes nominative like any other structurally case-marked nominal, as shown in (15).

(15) Taroo ga (Ken ni) hasiraserareta.
Taroo NOM Ken by run.CAUS.PASS.PAST
'Taroo was made to run (by Ken).'

There are several possible solutions. One is to add additional case tiers corresponding to each functional head, allowing each to restrict the case of the first K child of that head. So, a new *causative tier* would determine the case of causee, leaving the case of the causer up to the next higher tier. While this is technically possible, a more elegant solution makes use of the context-sensitive nature of Graf's (2018) tier projection and daughter string functions in the definition of the verbal case tier. We select the *highest v* head in each clause, that is, the one selected by T, increasing the context to a window of height 2. Then, we let the identity of the *v* head determine its daughter string language.

The indirect passive (adversative passive) can be handled in the same manner. Examples of this construction are given in (16) below.

(16) a. Ken ga Taroo ni hasirareta.
Ken NOM Taroo DAT run.PASS.PAST
'Ken was annoyed by Taroo running.'

b. Ken ga Taroo ni piano o
Ken NOM Taroo DAT piano ACC
hikareta.
play.PASS.PAST
'Ken was annoyed by Taroo playing the piano.'

Unlike in the causative construction, the embedded subject always receives dative case. We define the daughter string language of the indirect passive head accordingly. The revised definitions for both the verbal and genitive tiers are given in (17) below. A comparison of the old and new verbal case tier is shown in Figure 6.

(17) **Verbal case tier (revised)**
Project categories:
{*v*[-stat]/CAUS/IPASS daughter of T, K, C[–ECM]}
Daughter string languages:

v: $\{\text{NOM}, \text{GEN}, \text{LD}\} \cdot (\text{DAT}^* \cdot \{\text{ACC}, \text{LD}\})$

CAUS: $\{\text{NOM}, \text{GEN}\} \cdot \left\{ \begin{matrix} \{\text{ACC}, \text{DAT}\} \\ \text{DAT}^* \cdot \{\text{ACC}, \text{LD}\} \end{matrix} \right\}$

IPASS: $\{\text{NOM}, \text{GEN}\} \cdot \left\{ \begin{matrix} \text{DAT} \\ \text{DAT}^* \cdot \{\text{ACC}, \text{LD}\} \end{matrix} \right\}$

K/C: $\overline{\{\text{ACC}, \text{DAT}\}}^*$

(18) **Genitive case tier (revised)**
Project categories: {N, K, C[–ECM]}
Daughter string languages:
N: $\{\text{GEN}, \text{N}, \text{C}\}^*$
K/C: $\overline{\{\text{GEN}\}}^*$

4.8 Adjuncts

Adjuncts such as adverbs and PPs interfere with the tier constraints as currently defined, since there is in principle no bound to the number that may appear in any given case domain. We would like to ignore them since their presence does not affect case licensing. However, we cannot omit them on the tree tiers because any K heads they dominate would be interspersed among the daughters of a higher head. Instead, we must modify the daughter string languages, converting them from SL to TSL languages and ignoring adjuncts at this stage.

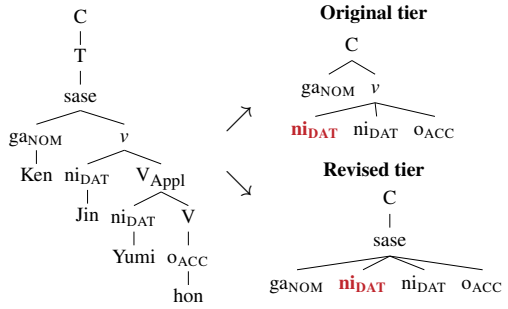


Figure 6: Original verbal case tier (top right) and revised version (bottom right) for causative ditransitive. In the original analysis, the causee (Jin) is the first K child of *v*, where it cannot be dative. In the revised version, all verbal arguments are daughters of *sase*, allowing the causee to be assigned dative in the usual manner.

To see why this works, recall again the form of the daughter string language of *v* on the verbal case tier, which (simplified) has the form $a(b^*c)$. Representing adjuncts by the symbol x and allowing them to occur anywhere, our string language is now $x^*a((x^*b)^*x^*c)x^*$. If we project a tier omitting x then the tier language is once again $a(b^*c)$.

While I cannot go into detail for reasons of space, the approach I have in mind makes use of *category-preserving selection* by means of adjunctizer heads. For example, the adjunctizer head for adjectives selects for categories A and N and itself is an N. The number of such heads in any MG lexicon is finite. We add all such heads to all *tree* tier projection functions, but omit them from them the daughter string tier projection function as just described.

4.9 Coordination

Due to the complexity of the data and the number of theories of coordination, it is beyond the scope of this paper to consider these in any depth, but I wish to at least outline the general problem and what it means for the analysis. Essentially, coordination is a problem when it splits a case domain, such as when a *v*P contains a coordinated VP. When we project the verbal case tier, the children of both Vs all end up as daughters of *v*; this predicts a change in case marking, which is contrary to fact.

Does this issue actually arise in Japanese? Perhaps not. Japanese allows coordination of TP and *v*P but not VP, and subjects may optionally remain in situ (cf. Hirata, 2006, and references therein). This means that whether each verb phrase has its own subject (remaining in situ), or both share a single subject (raised via across-the-board movement), there is no conflict. In a language similar where VP coordination is possible, we would need to restructure the analysis to include additional nested case domains (we avoided this earlier for the passive and causative constructions by using structure-sensitive tier projection). Should this prove unfeasible, this seems to be the most likely way in which the tree tier-based analysis could be invalidated.

It is at this point that I should note an alternative generalization of TSL to trees based on so-called *c[ommand]-strings* which, roughly speaking, encode chains of c-commanding elements (Graf and Shafiei, 2019). Because the present analysis already operates by collecting nominals in the daughter string of the case licensing domain node, it should be straightforward to recast it in terms of c-strings. This new version would also be robust against the domain-splitting problem presented by coordination. I leave the investigation of this possibility to future work.

5 Conclusion

In this paper, I developed a TSL analysis of Japanese case, and showed that the descriptive generalizations are captured neatly with a system of three tiers and a small number of constraints, and that the analysis extends with minimal modification to a wide range of constructions. The analysis is simple in computational terms and concise as a description of the case patterns of Japanese. These results support the proposal that the syntactic distribution of morphological case is TSL.

As mentioned earlier, the case patterns discussed in this paper also have close parallels in other languages. In particular, ergative case as analyzed in dependent case theory fits neatly into the current system, as does variation in case marking according to tense or aspect. It seems likely that this type of computational analysis can bring insight into our understanding of case marking across languages.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. BCS-1845344. I thank Thomas Graf for his feedback throughout this project. I also thank three anonymous reviewers for their comments. Reviewer 3 in particular provided numerous suggestions that improved the readability of the final paper.

References

Alëna Aksënova and Sanket Deshmukh. 2018. Formal restrictions on multiple tiers. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 64–73.

Mark Baker and Nadya Vinokurova. 2010. Two modalities of case assignment: Case in Sakha. *Natural Language & Linguistic Theory*, 28(3):593–642.

Shin Fukuda. 2007. Object case and event type: Accusative-dative object case alternation in Japanese. In *Annual Meeting of the Berkeley Linguistics Society*, volume 33, pages 165–176.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. *Proceedings of CLS*, 53:117–136.

Thomas Graf. 2022. Typological implications of tier-based strictly local movement. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2022*, pages 184–193.

Thomas Graf and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist Grammars. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*, pages 275–285.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 205–215.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological Typology, Phonetics and Phonology*, pages 126–195.

Ken Hiraiwa. 2001. On nominative-genitive conversion. *MIT Working Papers in Linguistics*, 39:66–125.

Ichiro Hirata. 2006. Predicate coordination and clause structure in Japanese. *Linguistic Review*, 23(1).

Tomoko Ishizuka. 2017. The passive voice. In *Handbook of Japanese Syntax*, pages 403–446. De Gruyter Mouton.

Hideki Kishimoto. 2018. On exceptional case marking phenomena in Japanese. *Kobe Papers in Linguistics*, 11:31–49.

Dakotah Lambert, Jonathan Rawski, and Jeffrey Heinz. 2021. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling*, 9(1):151–194.

Magdalena Lohninger, Iva Kovač, and Susanne Wurmbrand. 2022. From Prolepsis to Hyperraising. *Philosophies*, 7(2).

Hideki Maki and Asako Uchibori. 2008. Ga/no conversion. In *The Oxford handbook of Japanese linguistics*, pages 192–216. Oxford University Press.

Alec Marantz. 2000. Case and licensing. In *Arguments and case: Explaining Burzio's generalization*, pages 11–30. John Benjamins.

Shigeru Miyagawa. 1989. *Structure and Case Marking in Japanese*. Academic Press.

Shigeru Miyagawa and Takae Tsujioka. 2004. Argument structure and ditransitive verbs in Japanese. *Journal of East Asian Linguistics*, 13(1):1–38.

Edward P. Stabler. 1997. Derivational minimalism. In Christian Retore, editor, *Logical Aspects of Computational Linguistics*. Springer.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 267–276.

An Algebraic Characterization of Total Input Strictly Local Functions

Dakotah Lambert
Université Jean Monnet Saint-Étienne, CNRS
Institut d Optique Graduate School
Laboratoire Hubert Curien UMR 5516
F-42023, Saint-Étienne, France
dakotahlambert@acm.org

Jeffrey Heinz
Stony Brook University
Department of Linguistics
Institute for Advanced Computational Science
jeffrey.heinz@stonybrook.edu

Abstract

This paper provides an algebraic characterization of the total input strictly local functions. Simultaneous, noniterative rules of the form A→B/C__D, common in phonology, are definable as functions in this class whenever CAD represents a finite set of strings. The algebraic characterization highlights a fundamental connection between input strictly local functions and the simple class of definite string languages, as well as connections to string functions studied in the computer science literature, the definite functions and local functions. No effective decision procedure for the input strictly local maps was previously available, but one arises directly from this characterization. This work also shows that, unlike the full class, a restricted subclass is closed under composition. Additionally, some products are defined which may yield new factorization methods.

1 Introduction

The strictly local languages are those in which membership is decidable by the substrings up to some fixed width k of its words (McNaughton and Papert, 1971; Rogers and Pullum, 2011). Such languages are useful in the description of phonotactic patterns. Edlefsen et al. (2008) demonstrated that 75% of the patterns in the StressTyp2 database of stress patterns (Goedemans et al., 2015) are strictly local for k less than or equal to 6, reminiscent of Miller's Law on working memory, that an average person can hold roughly seven plus or minus two objects in short-term working memory (Miller, 1956). Even $k \leqslant 3$ suffices to capture nearly half of the patterns (see also Rogers and Lambert, 2019).

Chandlee et al. (2014) define the input strictly local functions to extend this notion to maps. They provide an efficient learner identifying functions in this class in the limit from positive data alone, using polynomial time and space. These mappings describe phonologically natural processes in which the output associated with a particular input symbol is uniquely determined by some local context around that symbol. Evidencing this naturality, 95 percent of maps in the P-Base database of phonological patterns (Mielke, 2008) lie in this class (Chandlee and Heinz, 2018). Related to this are the output strictly local maps, in which the output contributed by an input symbol is determined by the most recent symbols in the previous output (Chandlee et al., 2015).

One aspect of the study of formal languages is a deep connection between logic, automata, and algebra (Pin, 1997). Many classes of formal languages are characterized by decidable properties of an algebraic structure associated with each language in the class. The connection between algebraic structures and string languages can be extended to string-to-string maps based on the transducers that generate them (Filiot et al., 2016; Lambert, 2022).

This paper is structured as follows. Deterministic (sometimes called "unambiguous") finite-state acceptors (DFA) and transducers as well as the algebraic structures they induce are described in section 2. The formal definition of input strictly local maps is provided in section 3. The primary result, an algebraic characterization of this class, is given there alongside the polytime decision algorithm that it induces. This section also draws the connection to research in computer science which studied

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 25-34.
Amherst, Massachusetts, June 15-17, 2023

these functions under different names. Next in section 4 we discuss closure properties and an algorithm for composing transducers. We demonstrate that input strictly local functions are not closed under composition, but a subclass of them is so closed. Other operations under which the full class, perhaps with extensions, is closed are discussed in section 5. We conclude with discussion of these results in section 6.

2 Structures and Machines

A **semigroup** is a set S closed under some binary operation $\cdot$ (often denoted by adjacency) which is associative: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$. Given some finite alphabet Σ, the set of all nonempty sequences made up of those letters forms a semigroup with the concatenation operation. This is the **free semigroup** generated by Σ. A **monoid** is a semigroup in which there exists some element e such that for all x, $e \cdot x = x \cdot e = x$. Typically this identity element is represented by 1. The free semigroup generated by Σ can be adapted to the **free monoid** generated by Σ by including the empty sequence (denoted by λ), the identity for concatenation. The free semigroup and free monoid generated by Σ are often denoted Σ^+ and Σ^*, respectively.

A formal language L over Σ is some subset of this free monoid. Two useful equivalence relations can be defined based on L. **Nerode equivalence** is defined such that $a \overset{N}{\sim} b$ iff for all $v \in \Sigma^*$ it holds that $av \in L \Leftrightarrow bv \in L$ (Nerode, 1958). This is often called the Myhill-Nerode equivalence relation, as the well-known Myhill-Nerode theorem states that a language is regular iff its set of equivalence classes is finite. However, Myhill used a finer partition to achieve the same result: **Myhill equivalence** is defined such that $a \overset{M}{\sim} b$ iff for all $u \in \Sigma^*$, $ua \overset{N}{\sim} ub$; alternatively, for all $u, v \in \Sigma^*$, $uav \in L \Leftrightarrow ubv \in L$ (Rabin and Scott, 1959). Being a coarser partition, $\overset{N}{\sim}$ can never define more classes than $\overset{M}{\sim}$, and the number of classes defined by $\overset{M}{\sim}$ is in the worst case exponential in that defined by $\overset{N}{\sim}$ (Holzer and König, 2004), so finiteness in one translates to the other.

2.1 Illustrating Nerode and Myhill Relations

Consider the example language over $\{a, b, c\}$ consisting of all and only those words that do not contain an ab substring. Consider which classes must exist. $[\![ab]\!]$ is the set of words containing an ab substring. These are rejected, and no suffix can save them. So if $x, y \in [\![ab]\!]$, for all v it holds that $xv \notin L$ and $yv \notin L$. All of these words are related, and distinct from any accepted words. But the accepted words partition into two classes: words that end in a ($[\![a]\!]$) and others ($[\![\lambda]\!]$). The former are rejected after adding a b suffix, while the latter remain accepted after adding a b suffix. No suffix distinguishes words within these classes, so the three can define a minimal DFA for L (Nerode, 1958), as will be discussed shortly.

There are then at least three Myhill classes. But some classes split. An a prefix distinguishes the strings a and ba, and this generalizes. The class of accepted words ending in a splits to two: words ending in a that begin with b ($[\![ba]\!]$) and other words ending in a ($[\![a]\!]$). The other class of accepted words splits to three: words beginning in b ($[\![b]\!]$), nonempty words not beginning in b ($[\![c]\!]$), and the empty word ($[\![\lambda]\!]$). An a prefix distinguishes the first of these from the other two, while the $a\_\_b$ circumfix distinguishes the last from $[\![c]\!]$. The six $\overset{M}{\sim}$ classes are $[\![\lambda]\!]$, $[\![a]\!]$, $[\![b]\!]$, $[\![c]\!]$, $[\![ba]\!]$, and $[\![ab]\!]$.

The $\overset{N}{\sim}$ classes may have ill-defined concatenation. If $u \overset{N}{\sim} u'$, then $uv \overset{N}{\sim} u'v$ (it is a left congruence), but it may be that $v \overset{N}{\sim} v'$ while $uv \overset{N}{\not\sim} uv'$ (it is not a right congruence). In the current example, $b \overset{N}{\sim} c$ but $ab \overset{N}{\not\sim} ac$. In contrast, $\overset{M}{\sim}$ is compatible with concatenation (it is a congruence): if $u \overset{M}{\sim} u'$ and $v \overset{M}{\sim} v'$, it follows that $uv \overset{M}{\sim} u'v'$. That means these equivalence classes form the elements of a submonoid of Σ^*. The quotient monoid $\Sigma^*/\overset{M}{\sim}$ (these equivalence classes under concatenation) is the **syntactic monoid** of L. If L is a regular language, then this is the smallest monoid which can be used as a DFA accepting L (Rabin and Scott, 1959). The **syntactic semigroup** is $\Sigma^+/\overset{M}{\sim}$.

A string language is rational if and only if it has finitely many Myhill classes (Rabin and Scott, 1959). A **variety** of finite semigroups is a class closed under subsemigroup, quotients and finitary direct product (Pin, 1997). This implies closure under Boolean operations. Eilenberg's theorem states that these varieties uniquely define subclasses of rational languages (Eilenberg and Schützenberger, 1976). As we explain later they can also characterize subclasses of rational functions, such as the input strictly local functions.

2.2 String Acceptors

A DFA is a five-tuple $\langle \Sigma, Q, \delta, q_0, F \rangle$ where Σ is a finite alphabet, Q a finite state set, $\delta : \Sigma \times Q \to Q$

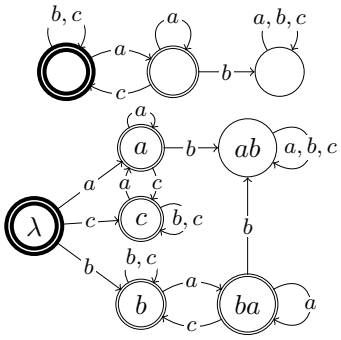


Figure 1: A DFA forbidding ab substrings induced by $\overset{N}{\sim}$ (above) and $\overset{M}{\sim}$ (below). States are labeled by class representatives. Doubly circled states are accepting and extra thick borders designate initial states.

a transition function, q_0 an initial state, and F a set of accepting states. A word is read one symbol at a time. If computation is in state q, the remaining string is σw, and $\delta(\sigma, q) = r$, then after one step, the computation will be in state r with remaining string w. Given the equivalence classes under $\overset{N}{\sim}$ or $\overset{M}{\sim}$, we can construct such an acceptor. Σ is the alphabet, Q the set of equivalence classes, $\delta(\sigma, q)$ the equivalence class of $q\sigma$, q_0 whichever class contains the empty sequence, and F the set of equivalence classes containing accepted words. Hopcroft and Ullman (1979) discuss a dynamic programming algorithm to reduce an arbitrary DFA to that induced by $\overset{N}{\sim}$. Another procedure, which will be described later, derives the Myhill relation from this form.

Figure 1 shows the acceptors induced by $\overset{N}{\sim}$ and $\overset{M}{\sim}$ for the example language over $\{a, b, c\}$ in which no word contains an ab substring. That induced by $\overset{M}{\sim}$ is a right Cayley graph of the syntactic monoid (see Zelinka, 1981), augmented with information about whether classes are accepting.

2.3 String-to-String Transducers

Oncina et al. (1993) discuss one method of generalizing these acceptors into functions. A **sequential transducer** is a five-tuple $\langle \Sigma, \Delta, Q, \delta, q_0 \rangle$, where Σ is the alphabet of the input, Δ that of the output, Q a finite set of states, $\delta : \Sigma \times Q \to \Delta^* \times Q$ a transition function, and q_0 an initial state. This behaves like an acceptor, where all strings in the domain are accepted and every edge traversed appends to an accumulating output. Sequential functions are total. A **subsequential transducer** generalizes this by associating outputs with states (Oncina et al., 1993); if an input word ends in state q, the output receives the suffix associated with q. The function σ mapping states to suffixes is added as a sixth element: $\langle \Sigma, \Delta, Q, \delta, q_0, \sigma \rangle$. The names and order of these components here are not the same as those used in the original work, but seem to have become commonly used in later work. Adding another element, a string prefixed to all output strings, adds nothing because it could be added to each edge out of q_0 and to that state's output. So in this work, this universal prefix π will be assumed: $\langle \Sigma, \Delta, Q, \delta, q_0, \pi, \sigma \rangle$. This change leaves most definitions unaffected.

Bruyère and Reutenauer (1999) argue that the subsequential notion is more deserving of the status as the basic object, and refer to such functions as simply sequential, a practice followed by Lombardy and Sakarovitch (2006), among others. A subsequential machine is equivalent to a sequential machine over a larger alphabet that includes explicit boundary symbols, and a well-formed version of the latter can be rewritten as the former. Given this bijection, the remainder of this work will follow this recent notational trend.

Sequentiality may depend on the direction in which the input is read. Iterative regressive harmony patterns cannot be described by left-to-right sequential functions as they admit unbounded delay between seeing a harmonizing symbol and finding the trigger that determines its surface form (Heinz and Lai, 2013; Mohri, 1997). However, this process can be expressed as a right-to-left sequential function. This is equivalent to reversing the output of a left-to-right transducer applied to the input's reversal. Or one could say the machine reads the string from right to left, prefixing to the output. If SQ is the sequential class, we denote the left-to-right class $\to$SQ and its right-to-left variant $\leftarrow$SQ, with the arrow indicating directionality.

The **longest common prefix** (denoted lcp) of a set of strings S is the unique string u such that u is a prefix of every string in S and that u is longer than every other string u' which prefixes every string in S. A transducer is **onward** if it emits output as early as it can: for all states p, $\mathrm{lcp}(\{y \in \Delta^* : \delta(a, p) = \langle y, q \rangle\} \cup \{\sigma(p)\}) = \lambda$. The Nerode equivalence relation extends naturally to functions by means of the **tails** of input strings. The set of tails of x in a function f, $T_f(x)$, is defined as follows:

$$T_f(x) = \{\langle y, v \rangle : f(xy) = \mathrm{lcp}(f(x\Sigma^*))v\}.$$

Two strings are related iff their tails are equal. We write this relation as $\overset{N}{\sim}$, emphasizing connection to

Nerode equivalence for string sets. A transducer in canonical form is onward and has one state per $\overset{N}{\sim}$ class. A two-sided extension generalizes Myhill equivalence. The **contexts** of x in f are as follows:

$$C_f(x) = \{\langle w, y, v\rangle \colon f(wxy) = \mathrm{lcp}(f(wx\Sigma^*))v\}.$$

The subset where $w = \lambda$ is essentially equivalent to $T_f(x)$, so the $\overset{M}{\sim}$ relation derived from C forms, as with string sets, a refinement of $\overset{N}{\sim}$.

2.4 Monoids from Canonical Machines

The canonical form of a machine derives states from the $\overset{N}{\sim}$ relation. The $\overset{M}{\sim}$ relation accounts for the influence of prefixes. So, to construct a machine over $\overset{M}{\sim}$ from a canonical machine, i.e. to construct a right Cayley graph of its associated monoid, we look to see where each input symbol takes each of the states. In other words, what is the action of each symbol over the states? This is the transition congruence (Filiot et al., 2016). McNaughton and Papert (1971) use this same construction.

Consider the automata of Figure 1. Assign an arbitrary number to each state of the automaton induced by $\overset{N}{\sim}$: $[\![\lambda]\!]$ is 1, $[\![a]\!]$ is 2, and $[\![ab]\!]$ is 3. Denote by $\langle x, y, z\rangle$ the function mapping 1 to x, 2 to y, and 3 to z. The identity function, $\langle 1, 2, 3\rangle$, corresponds to λ. From there, a, b, and c act as $\langle 2, 2, 3\rangle$, $\langle 1, 3, 3\rangle$ and $\langle 1, 1, 3\rangle$, respectively. The complete structure extends from these. Consider ab: this first applies the a mapping, then applies that of b to its result. So $\langle 1, 2, 3\rangle$ maps first to $\langle 2, 2, 3\rangle$ by a then to $\langle 3, 3, 3\rangle$ by b. By the same process, we find that $aa = ca = a$, $ac = cb = cc = c$, $bb = bc = b$ and ba is a new state $\langle 2, 3, 3\rangle$. Extending $ab = \langle 3, 3, 3\rangle$ and $ba = \langle 2, 3, 3\rangle$, we find $ab{\cdot}a = ab{\cdot}b = ab{\cdot}c = ba{\cdot}b = ab$, $ba{\cdot}a = ba$ and finally $ba \cdot c = b$. Iteration generated no new states, so the process is complete. This conforms to the structure shown in Figure 1, whose Cayley graph is shown at the top in Table 1.

Note that the complement of the language forbidding ab substrings – the language of words with ab substrings – shares the same syntactic semigroup. This holds in general: an automaton and its complement share the same algebraic structure, as state parity is independent from the actions of transitions. It follows that classes defined purely by semigroup properties must be closed under complement.

Now consider the transducer of Figure 2. This transducer is a representation of intervocalic voicing, a phonological process where voiceless obstruents become voiced between vowels. As a phonological rule this is T→D/V__V. For example, this transducer maps the string TVTVD to TVDVD.

	a	b	c	ab	ba
a	a	ab	c	ab	ab
b	ba	b	b	ab	ba
c	a	c	c	ab	a
ab	ab	ab	ab	ab	ab
ba	ba	ab	b	ab	ab

	T	V	D	VT
T	D	V	D	VT
V	VT	V	D	VT
D	D	V	D	VT
VT	D	V	D	VT

Table 1: The Cayley table for the syntactic semigroups in Figure 1 (above) and Figure 2 (below).

The transducer above is in canonical form, where each state represents one $\overset{N}{\sim}$ class. State 2 is all those strings that end in V, state 3 those ending in VT, and state 1 all others. The five actions are the identity $\langle 1, 2, 3\rangle$ corresponding to λ, $\langle 1, 1, 1\rangle$, $\langle 1, 3, 1\rangle$, and $\langle 2, 2, 2\rangle$ corresponding to D, T, and V, respectively, and finally $\langle 3, 3, 3\rangle$ for VT. One can verify that for each class, some context distinguishes its words from words in each other class, and that no context distinguishes words within a class. For example, a V__V context separates λ and T, as for λ the following V contributes V alone while for T it contributes DV. Technically, $\langle \mathrm{V}, \mathrm{V}, \mathrm{V}\rangle \in C(\lambda)$ while $\langle \mathrm{V}, \mathrm{V}, \mathrm{DV}\rangle \in C(\mathrm{T})$, but by determinism the triples are unique in their first two components. A VT__λ context separates λ and D, as the λ contributes T to the former but λ to the latter. That no context distinguishes strings within a class is guaranteed by the construction. The Cayley graph corresponding to the monoid in Figure 2 is shown at bottom in Table 1.

This construction appears to discard output information, but it is recoverable. Outputs may be compatibly assigned to the states and edges and the result used as a transducer. Its structure is the same as that of the string language in which all words end in "VT". This notion of structural equivalence gives rise to a deep theory of function complexity.

2.5 Definite Algebraic Structure

A string language L is **definite** if can be defined by a finite set X of permitted suffixes: $L = \{wv :$

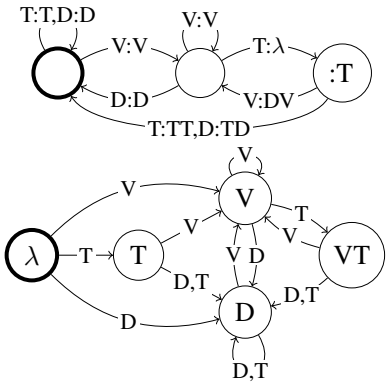


Figure 2: Transducer and monoid for "T becomes D directly between two V".

$w \in \Sigma^*, v \in X\}$ (Perles et al., 1963). The class of definite languages is denoted $\mathcal{D}$. Because X is a finite set it holds some longest string of length n. Whether a string belongs to L can be decided by examining its last n symbols. Such languages are called n-definite. More generally, as the canonical acceptor for a definite language processes strings, the states correspond to strings in Σ^n representing the most recent history. In the sense of Jurafsky and Martin (2008) then, the state space of definite languages is Markovian.

The definite languages were one of the early classes of formal languages to be given an algebraic characterization (Brzozowski and Simon, 1973; Brzozowski and Fich, 1984). Many algebraic structures are defined in terms of idempotents. An element e of a monoid is **idempotent** if $e \cdot e = e$. As an example, the idempotents of the syntactic semigroups shown in 1 are $\{a, b, c, ab\}$ and $\{\mathrm{V}, \mathrm{D}, \mathrm{VT}\}$, respectively. Denote by E the set of idempotents.

An algebraic property characterizes exactly the definite languages (Brzozowski and Simon, 1973; Brzozowski and Fich, 1984). Syntactic semigroups of definite languages have the property that for all $e \in E$, $x \in S$, it holds that $xe = e$, often written $Se = e$ with universal quantification left implicit.

The string language which forbids ab substrings is not definite. This follows from the algebraic characterization and from the Cayley table for this language in Table 1. While b is an idempotent (since $b \cdot b = b$), $a \cdot b = ab \neq b$. Thus $Se \neq e$.

For intervocalic voicing it holds that $Se = e$ for all its idempotents $e \in \{\mathrm{V}, \mathrm{D}, \mathrm{VT}\}$. One verifies this by examining their columns in the Cayley table in Table 1. As its minimal transducer processes input, the most recently read symbols fix its state.

The syntactic semigroups such that $Se = e$ form the variety $\mathcal{D}$ (Brzozowski and Simon, 1973; Brzozowski and Fich, 1984). It follows they are closed under subsemigroup, quotients, finite direct products, and thus the Boolean operations. This variety has played a key role in developing an algebraic theory of recognizable languages (Straubing, 1985).

3 Input Strictly Local Functions

Chandlee et al. (2014) define input strictly local transducers by a restriction on the tails, inducing a canonical structure. A function is input strictly local iff for some natural number k, the function is definable by a sequential transducer whose states are labeled by $\Sigma^{<k}$, q_0 is the state labeled by λ, and edges are of the form $\delta(a, q) = \langle w, \mathrm{Suff}^{k-1}(qa)\rangle$. The suffix function is defined as expected:

$$\mathrm{Suff}^n(w) = \begin{cases} \lambda & \text{if } n \leqslant 0, \\ w & \text{if } |w| \leqslant n, \\ v & \text{if } w = uv \text{ for } u \in \Sigma^*, v \in \Sigma^n. \end{cases}$$

This canonical form is a monoid. The operation $u \cdot v = \mathrm{Suff}^{k-1}(u \cdot v)$ is associative, and λ is the identity. Let f be a function, $\overrightarrow{S}$ and $\overleftarrow{S}$ be the semigroups of the left-to-right and right-to-left transducers associated with f, respectively, and e range over idempotents of the appropriate semigroup.

Theorem 1. *The following are equivalent:*

- *f is a total input strictly local function*
- *f is $\to \mathcal{D}$:* $\overrightarrow{S}e = e$
- *f is $\leftarrow \mathcal{D}$:* $\overleftarrow{S}e = e$

Proof. The nonidentity idempotent elements of this monoid are Σ^{k-1}, as if $x \in \Sigma^{k-1}$ we have $x = \mathrm{Suff}^{k-1}(x) = \mathrm{Suff}^{k-1}(xx)$ and if $x \in \Sigma^{<k-1}$ we instead have $x \neq \mathrm{Suff}^{k-1}(xx)$. If $x \in \Sigma^{k-1}$ we have that $\mathrm{Suff}^{k-1}(ux) = x$ for all $u \in \Sigma^*$, so for all elements s it holds that $s \cdot x = x$. In other words, $Se = e$ for all idempotent elements e in the syntactic semigroup (which excludes the identity). This is the property characterizing definite languages, defined by a set of permitted suffixes (Brzozowski and Simon, 1973; Brzozowski and Fich, 1984).

The directionality statement follows from the fact that input strictly local functions are not directional (Chandlee and Heinz, 2018). □

The canonical form of an input strictly local transducer is the same as that of a definite string

language (Perles et al., 1963). Both are defined by the k most recent symbols encountered fixing the state, with no long-distance effects. Indeed, this class has been discussed as the definite (Krohn et al., 1967; Stiffler, 1973) or local (Vaysse, 1986) functions decades before Chandlee et al. (2014) introduced them to linguists as input strictly local.

We invoke this characterization of input strictly local functions as definite structures to provide an effective decision procedure for the class. First, it is converted to a canonical sequential form by the algorithm of Mohri (1997). If conversion fails, the map is certainly not in the class, as it is not even sequential. Otherwise, the syntactic semigroup is constructible by the algorithms shown in section 2 (McNaughton and Papert, 1971). Finally one needs only to check that for each idempotent e and each element s, $se = e$. Recall that the identity is in the semigroup iff it is reachable by a nonempty string.

Strictly local string languages follow the same structure but additionally allow transition to a rejecting sink in lieu of some otherwise expected transitions. These changes do not necessarily retain the algebraic structure, but a semigroup can be regenerated by the usual method. Accounting for whether a factor in some fixed set has ever occurred admits some long-distance dependency.

4 Composing Functions

Closure properties provide important insight into classes of languages. An intersection-closed class admits new patterns satisfying its properties defined by coöccurrence of patterns in that class. Many subregular classes are so closed, and learning a strictly piecewise pattern as a coöccurrence of constraints has proven more effective than learning a single pattern (Heinz and Rogers, 2013). (Pseudo)varieties of finite semigroups are closed under finitary products, subsemigroups, and quotients (Eilenberg and Schützenberger, 1976). Intersections and unions of automata are computed from a product, extracting the reachable subsemigroup and minimizing the result by a quotient. Automata share structure with their complements, so varieties define classes closed under Boolean operations. The property defining definite languages, that $Se = e$ for all idempotents e, yields a variety, $\mathbf{D}$, of finite semigroups. These languages are then Boolean-closed.

If coöccurring factors are a basic unit of string languages, composed rules might be a basic unit of

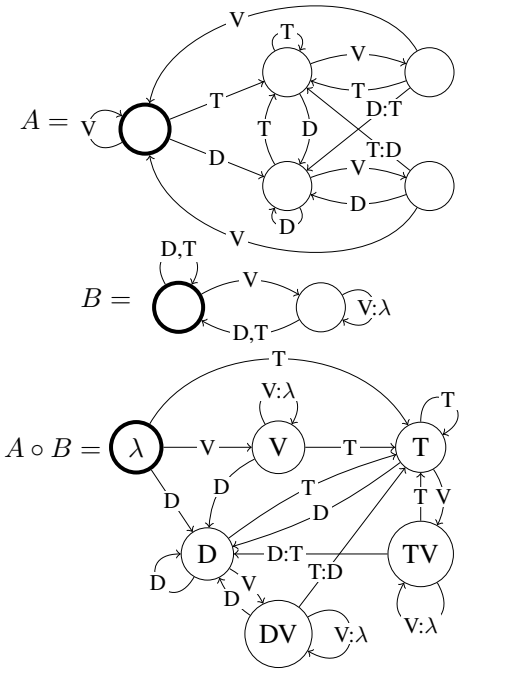


Figure 3: ISL is not composition-closed.

functions. Let δ^* denote the transitive closure of δ:

$$\delta^*(w,x) = \begin{cases} \langle \lambda, x \rangle & w = \lambda \\ \langle uv, y \rangle & w = aw', a \in \Sigma, \\ & \langle u, x' \rangle = \delta(a, x), \\ & \langle v, y \rangle = \delta^*(w', x') \end{cases}$$

Then if $f = \langle \Delta, \Gamma, Q_f, \delta_f, q_{0f}, \pi_f, \sigma_f \rangle$ and $g = \langle \Sigma, \Delta, Q_g, \delta_g, q_{0g}, \pi_g, \sigma_g \rangle$, the composition $f \circ g$ computes the result of applying f to the output of g. This composition is effectively constructible (Mohri, 1997). A construction is as follows:

$$\begin{aligned} f \circ g &= \langle \Sigma, \Gamma, Q_f \times Q_g, \delta_\circ, q_i, \pi_f \alpha, \sigma_\circ \rangle \\ \langle \alpha, r \rangle &= \delta_f^*(\pi_g, q_{0f}) \\ q_i &= \langle r, q_{0g} \rangle \\ \delta_\circ(a, \langle m, n \rangle) &= \Big\{ \langle w, \langle s, t \rangle \rangle : \delta_g(a, n) = \langle u, t \rangle, \\ &\qquad \delta_f^*(u, m) = \langle w, s \rangle \Big\} \\ \sigma_\circ(\langle m, n \rangle) &= \sigma_f(\delta_f^*(m, \sigma_g(n))_1) \end{aligned}$$

This composition is not a direct product in the algebraic sense. The state space is the product space, but the action is not the natural pointwise action defining the direct product. Thus, composition closure is not free and in general does not hold.

The transducers shown in Figure 3 exhibit this nonclosure for definite functions. The first, A, is

simultaneous application of two rules, D→T/TV__ and T→D/DV__, a voicing assimilation across a single vowel. Then B is a vowel-span truncation: V→∅/V__. By applying B and then A, the context in which T or D changes becomes unboundedly long. The strings V^n and DV^n have the same n-suffix for any n, but a suffixed T contributes a T to the first and a D to the latter. The two have differing tails, failing to satisfy input strict locality. In semigroup terms, V is idempotent as V and VV lie in the same class, but DVV is DV and not V. Thus $Se \neq e$ and the function is not definite. In fact, the resulting monoid is not even locally a semilattice (locally testable, Brzozowski and Simon, 1973) nor $\mathscr{J}$-trivial (piecewise testable, Simon, 1975). It is everywhere-idempotent, which in string languages would imply definability in two-variable first-order logic of general precedence alone (Brzozowski and Fich, 1984; Kufleitner and Weil, 2010).

One subclass of definite functions is composition closed: that where only bounded spans may delete.

Theorem 2. *If f and g are definite functions and if all input sequences of length k to g are guaranteed to produce nonempty output, then $f \circ g$ is definite.*

Proof. If f is m-definite, g is n-definite, and input sequences of g are guaranteed to contribute nonempty output after at most k symbols, then after mk input symbols, g must have produced at least m intermediate output symbols. This fixes the state in f. The state of g is fixed after n or more symbols. So the degree of definiteness of $f \circ g$ is at most the greater of n and mk. □

Corollary 1. *The subclass of definite functions deleting only bounded spans is composition closed.*

Proof. If f and g are definite and guarantee nonempty output after reading at most k and ℓ symbols, respectively, then $f \circ g$ yields nonempty output after reading at most ℓk symbols. The composition remains in this subclass. □

The machines of Figure 3 do not compose to a definite machine because unbounded spans of V delete, collapsing to just V no matter their length.

Many phonologically relevant patterns lie in this subclass, including some that optimality theory has struggled to analyze (Chandlee et al., 2018). Interconsonantal schwa deletion, intervocalic voicing, and word-final devoicing are each computed by transducers where all input sequences of length two contribute nonempty output. In this order, their composition is shown in Figure 4, and it is definite of degree four: every string of length three synchronizes the machine. Moreover all length four input sequences produce nonempty output.

Readers familiar with the literature on local functions may recall results that seem stronger than our Theorem 2. For example, Sakarovitch (2009, p. 664) states that if g is a proper local function of degree d and f a local function of degree d', then the composition $f \circ g$ is local of degree $d + d'$. In that work, a proper local function is one in which no deletion occurs. This is a more restrictive constraint than our own, as we allow for deletion in bounded spans. Similarly, Vaysse (1986, p. 168) states that the composition of any local function f of degree d and any local function g of degree d' is local of degree $d + d' - 1$. This, however, takes place in a richer setting in which transitions not only might append symbols to the output, but also might delete previous symbols. Neither previous result is directly applicable here.

5 Other Kinds of Operations

Although a subclass of the definite functions is closed under composition, the class as a whole is not. In general, function composition does not preserve algebraic properties. This section discusses a general type of machine that unifies transducers, DFAs, weighted automata, and more. Operations on these general automata that behave like products will preserve algebraic properties and allow complex systems to be factored in an algebraically natural way. Some such operations are shown here.

The outputs of a transducer influence its semigroup structure only by preventing state merges. Mohri (Lothaire, 2005) describes a more general notion of a transducer whose outputs are elements in some semiring rather than mere strings. Sequential transducers are input-deterministic, so the operation combining paths is unnecessary. We can think about machines whose output lies in some monoid. Standard transducers satisfy this property: if the output alphabet is Δ then the output monoid is Δ^* under concatenation.

Consider then a system in which the output monoid is not Δ^* but regular languages over Δ. The form of the output is irrelevant, but for concreteness suppose we are dealing with DFAs over Δ. A definite transducer may be translated directly into this form by replacing the output strings with a DFA accepting that string alone, with one state more

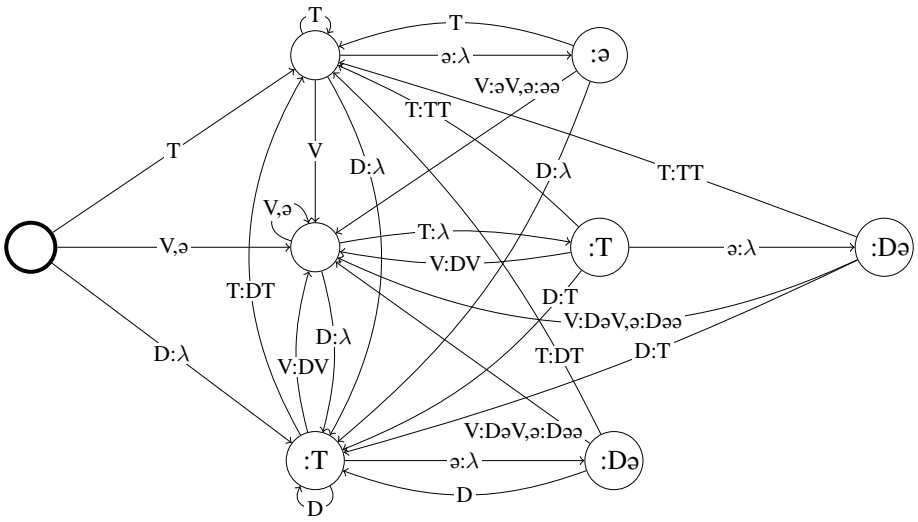


Figure 4: Interconsonantal schwa deletion, then intervocalic voicing, then word-final devoicing, all composed.

than the string's length. We define three distinct products over this structure: pointwise evaluation, union, and parallel application with preference. In the following discussion, machines are defined with an output monoid in place of an output alphabet.

If we have $f = \langle \Sigma, \Delta^*, Q_f, \delta_f, q_{0f}, \pi_f, \sigma_f \rangle$ and $g = \langle \Sigma, \Gamma^*, Q_g, \delta_g, q_{0g}, \pi_g, \sigma_g \rangle$, we define their pointwise evaluation product, $f \odot g$, as follows:

$$f \odot g = \big\langle \Sigma, \Delta^* \times \Gamma^*, Q_f \times Q_g, \delta_{f\odot g}, \\ \langle q_{0f}, q_{0g} \rangle, \langle \pi_f, \pi_g \rangle, \sigma_{f\odot g} \big\rangle,$$

where $\sigma_{f\odot g}(\langle q, r \rangle) = \langle \sigma_f(q), \sigma_g(r) \rangle$, pointwise application of suffixing, and if $\delta_f(a, q) = \langle u, q' \rangle$ and $\delta_g(a, r) = \langle v, r' \rangle$ then $\delta_{f\odot g}(a, \langle q, r \rangle) = \langle \langle u, v \rangle, \langle q', r' \rangle \rangle$. The operation is pointwise concatenation: $\langle a, b \rangle \cdot \langle c, d \rangle = \langle ac, bd \rangle$. The pair that $f \odot g$ derives from an input w juxtaposes the result of applying f to w or that of applying g to w.

Let $\mathscr{A}_X$ represent the DFAs over alphabet X. If we have $f = \langle \Sigma, \mathscr{A}_\Delta, Q_f, \delta_f, q_{0f}, \pi_f, \sigma_f \rangle$ and $g = \langle \Sigma, \mathscr{A}_\Gamma, Q_g, \delta_g, q_{0g}, \pi_g, \sigma_g \rangle$, we define the unioned product of f and g, $f \boxdot g$ as follows:

$$f \boxdot g = \big\langle \Sigma, \mathscr{A}_{\Delta\cup\Gamma}, Q_f \times Q_g, \delta_{f\boxdot g}, \\ \langle q_{0f}, q_{0g} \rangle, \{\pi_f, \pi_g\}, \sigma_{f\boxdot g} \big\rangle,$$

where $\sigma_{f\boxdot g}(\langle q, r \rangle) = \{\sigma_f(q), \sigma_g(r)\}$, the union of the outputs of the two suffixing functions, and if $\delta_f(a, q) = \langle u, q' \rangle$ and $\delta_g(a, r) = \langle v, r' \rangle$ then $\delta_{f\boxdot g}(a, \langle q, r \rangle) = \langle u \cup v, \langle q', r' \rangle \rangle$. Every input symbol admits choice, applying either f or g.

For homogeneous functions we have a final operation: apply both at once, outputting from the left machine if it changes the input, else from the right machine. Let $f = \langle \Sigma, \Sigma^*, Q_f, \delta_f, q_{0f}, \pi_f, \sigma_f \rangle$ and $g = \langle \Sigma, \Sigma^*, Q_g, \delta_g, q_{0g}, \pi_g, \sigma_g \rangle$, and define this change-preferring product as follows:

$$f \diamondsuit g = \big\langle \Sigma, \Sigma^*, Q_f \times Q_g, \delta_{f\diamondsuit g}, \\ \langle q_{0f}, q_{0g} \rangle, \pi_{f\diamondsuit g}, \sigma_{f\diamondsuit g} \big\rangle,$$

where $\pi_{f\diamondsuit g}$ is equal to π_f unless that is λ in which case it is equal to π_g, and similarly $\sigma_{f\diamondsuit g}(\langle q, r \rangle)$ is equal to $\sigma_f(q)$ unless that is λ in which case it is equal to $\sigma_g(r)$, and finally if $\delta_f(a, q) = \langle u, q' \rangle$ and $\delta_g(a, r) = \langle v, r' \rangle$ then $\delta_{f\diamondsuit g}(a, \langle q, r \rangle) = \langle w, \langle q', r' \rangle \rangle$, where $w = v$ if $u = a$ or else $w = u$. For two processes that do not affect one another, this is algebraic-property preserving composition.

These combinators are built on the product construction that Rabin and Scott (1959) and Hopcroft and Ullman (1979) use for unions or intersections of DFAs. The transition semigroup of the result is the product of those of the inputs. Definite machines are defined by a variety, and so are product closed, which means the $\odot$, $\boxdot$, and $\diamondsuit$ combinators yield definite machines from definite inputs.

Consider then that deletion of schwa between two consonants is definite, defined by the rule ə→∅/C__C. This is a definite function, by the construction used by Chandlee (2014). The identity function is also definite, having but a single state. Applying $\boxdot$ yields the (definite) deterministic rational relation of Figure 5 implementing optional interconsonantal schwa deletion. Some deterministic rational relations have been studied (Beros and de la Higuera, 2016), and this algebraic perspective

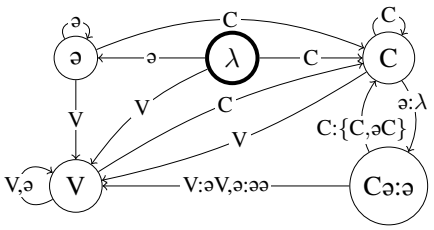


Figure 5: Optional schwa deletion between consonants.

offers a general mechanism for dealing with them.

6 Conclusion

Input strictly local maps suffice to describe a large portion of phonological processes. They are definite functions (Krohn et al., 1967; Stiffler, 1973). They are local functions (Vaysse, 1986; Sakarovitch, 2009). Given a minimal sequential finite-state transducer representing a mapping, we showed that it is decidable in time polynomial in the size of the transition semigroup of the machine whether the process is input strictly local: all idempotents must be right zeros. Using this characterization, we have shown that this class of functions cannot be closed under composition, but that this closure does hold for a restricted subclass in which deletion may occur only in bounded spans.

In these functions, only a local context around a symbol can influence its output. They do not exhibit the long-distance effects that strictly local string languages allow, where a single factor might cause computation to fall into a sink state for the remainder of the run. Definite languages are all strictly local, but so are, say, reverse definite languages. These have the opposite characterization, defined by semigroups where $eS = e$. These are the functions where $\text{Pref}^{k-1}(u) = \text{Pref}^{k-1}(v)$ implies $u \overset{N}{\sim} v$, that $T(u) = T(v)$.

Current research involves exploring the function analogues of some of the other classes that correspond to subregular string languages, such as these reverse definite functions, and classifying natural language patterns accordingly (Lambert, 2022). Additional lines of future research include better understanding how algebraic properties can fuel grammatical inference of string functions (de la Higuera, 2010), and the factorization of string functions into component parts along the lines of Rogers and Lambert (2019).

References

Achilles Beros and Colin de la Higuera. 2016. A canonical semi-deterministic transducer. *Fundamenta Informaticae*, 146(4):431–459.

Véronique Bruyère and Christophe Reutenauer. 1999. A proof of Choffrut's theorem on subsequential functions. *Theoretical Computer Science*, 215(1–2):329–335.

Janusz Antoni Brzozowski and Faith Ellen Fich. 1984. On generalized locally testable languages. *Discrete Mathematics*, 50:153–169.

Janusz Antoni Brzozowski and Imre Simon. 1973. Characterizations of locally testable events. *Discrete Mathematics*, 4(3):243–271.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language*, pages 112–125, Chicago, USA. Association for Computational Linguistics.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry*, 49(1):23–60.

Jane Chandlee, Jeffrey Heinz, and Adam Jardine. 2018. Input strictly local opaque maps. *Phonology*, 35(2):171–205.

Colin de la Higuera. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Matt Edlefsen, Dylan Leeman, Nathan Myers, Nathaniel Smith, Molly Visscher, and David Wellcome. 2008. Deciding strictly local (SL) languages. In *Proceedings of the 2008 Midstates Conference for Undergraduate Research in Computer Science and Mathematics*, pages 66–73.

Samuel Eilenberg and Marcel-Paul Schützenberger. 1976. On pseudovarieties. *Advances in Mathematics*, 19(3):413–418.

Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. 2016. First-order definability of rational transductions: An algebraic approach. In *LICS '16: Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 387–396. Association for Computing Machinery.

R. W. N. Goedemans, Jeffrey Heinz, and Harry van der Hulst. 2015. StressTyp2.

Jeffrey Heinz and Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language*, pages 52–63, Sofia, Bulgaria. Association for Computational Linguistics.

Jeffrey Heinz and James Rogers. 2013. Learning subregular classes of languages with factored deterministic automata. In *Proceedings of the 13th Meeting on the Mathematics of Language*, pages 64–71, Sofia, Bulgaria. Association for Computational Linguistics.

Markus Holzer and Barbara König. 2004. On deterministic finite automata and syntactic monoid size. *Theoretical Computer Science*, 327(3):319–347.

John Edward Hopcroft and Jeffrey David Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Daniel Jurafsky and James Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*, 2nd edition. Prentice-Hall, Upper Saddle River, NJ.

Kenneth Krohn, Richard Mateosian, and John Rhodes. 1967. Methods of the algebraic theory of machines: Decomposition theorem for generalized machines; Properties preserved under series and parallel compositions of machines. *Journal of Computer and System Sciences*, 1(1):55–85.

Manfred Kufleitner and Pascal Weil. 2010. On the lattice of sub-pseudovarieties of $\mathbf{DA}$. *Semigroup Forum*, 81:243–254.

Dakotah Lambert. 2022. *Unifying Classification Schemes for Languages and Processes with Attention to Locality and Relativizations Thereof*. Ph.D. thesis, Stony Brook University.

Silvain Lombardy and Jacques Sakarovitch. 2006. Sequential? *Theoretical Computer Science*, 356(1–2):224–244.

M. Lothaire. 2005. *Applied Combinatorics on Words*. Cambridge University Press, New York.

Robert McNaughton and Seymour Aubrey Papert. 1971. *Counter-Free Automata*. MIT Press.

Jeff Mielke. 2008. *The Emergence of Distinctive Features*. Oxford University Press, New York, NY.

George Armitage Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Anil Nerode. 1958. Linear automaton transformations. In *Proceedings of the American Mathematical Society*, volume 9, pages 541–544. American Mathematical Society.

José Oncina, Pedro García, and Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458.

Micha A. Perles, Michael Oser Rabin, and Eliahu Shamir. 1963. The theory of definite automata. *IEEE Transactions on Electronic Computers*, 12(3):233–243.

Jean-Éric Pin. 1997. Syntactic semigroups. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages: Volume 1 Word, Language, Grammar*, pages 679–746. Springer-Verlag, Berlin.

Michael Oser Rabin and Dana Scott. 1959. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125.

James Rogers and Dakotah Lambert. 2019. Extracting Subregular constraints from Regular stringsets. *Journal of Language Modelling*, 7(2):143–176.

James Rogers and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20(3):329–342.

Jacques Sakarovitch. 2009. *Elements of Automata Theory*. Cambridge University Press.

Imre Simon. 1975. Piecewise testable events. In Helmut Brakhage, editor, *Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 214–222. Springer-Verlag, Berlin.

Price Stiffler, Jr. 1973. Extension of the fundamental theorem of finite semigroups. *Advances in Mathematics*, 11(2):159–209.

Howard Straubing. 1985. Finite semigroup varieties of the form $V * D$. *Journal of Pure and Applied Algebra*, 36:53–94.

Odile Vaysse. 1986. Addition molle et fonctions p-locales. *Semigroup Forum*, 34:157–175.

Bohdan Zelinka. 1981. Graphs of semigroups. *Časopis pro pěstování matematiky*, 106(4):407–408.

Analogy in Contact: Modeling Maltese Plural Inflection

Sara Court
The Ohio State University
court.22@osu.edu

Andrea D. Sims
The Ohio State University
sims.120@osu.edu

Micha Elsner
The Ohio State University
elsner.14@osu.edu

Abstract

Maltese is often described as having a hybrid morphological system resulting from extensive contact between Semitic and Romance language varieties. Such a designation reflects an etymological divide as much as it does a larger tradition in the literature to consider concatenative and non-concatenative morphological patterns as distinct in the language architecture. Using a combination of computational modeling and information theoretic methods, we quantify the extent to which the phonology and etymology of a Maltese singular noun may predict the morphological process (affixal vs. templatic) as well as the specific plural allomorph (affix or template) relating a singular noun to its associated plural form(s) in the lexicon. The results indicate phonological pressures shape the organization of the Maltese lexicon with predictive power that extends beyond that of a word's etymology, in line with analogical theories of language change in contact.

1 Introduction

Maltese is a Semitic language that has been shaped by an extensive history of contact with non-Semitic languages. A large influx of Sicilian, Italian, and English words over the course of hundreds of years has influenced the Maltese lexicon and grammar, making it a prime case study for those interested in the effects of language contact on morphological systems. Semitic languages are notable for their use of root-and-pattern (a.k.a. templatic) morphology in which inflectional or derivational forms of a lexeme may be related via the non-concatenative interleaving of consonants and vowels. In Maltese, some lexemes of non-Semitic origin have integrated into the native morphology to take both concatenative as well as non-concatenative patterns of Semitic origin. Non-Semitic morphological markers have also entered the grammar and may be found on lexemes of both non-Semitic and Semitic origin.

This study applies methods from computational modeling and information theory to investigate factors shaping the organization of the modern Maltese lexicon. Contextualized within frameworks of analogical classification and usage-based accounts of contact-induced language change, we quantify the extent to which the phonology and etymology of Maltese lexemes are predictive of nominal plural inflection in the language. The results indicate that system-level phonology, hypothesized to capture analogical pressures, and etymology, hypothesized to capture conservative pressures that resist analogical change, are predictive of Maltese plural inflection in non-redundant ways, with phonology being more predictive than etymology overall.

Because Maltese is a Semitic language, we are also interested in the extent to which these factors are predictive of the type of morphology (either concatenative or non-concatenative) relating singular-plural pairs in the language. Our results show that both phonology and etymology are twice as predictive of a lexeme's plural allomorph(s) as compared to its concatenative type. This suggests that the analogical processes hypothesized to inform speakers' morphological intuitions are most sensitive to phonological similarities across surface forms, regardless of typological differences distinguishing concatenative and non-concatenative relationships. This study provides quantitative evidence for the role of analogical classification based on phonological similarity at the word level as a structuring principle of Maltese nominal plural morphology.

2 Morphology in Contact: Maltese as a "Hybrid" Language?

Maltese is a descendant of the Siculo Arabic variety spoken by settlers of the Maltese islands beginning in the year 1048 (Fabri, 2010; Brincat, 2011). While the language is Semitic with respect to its genetic classification, isolation and centuries of for-

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 35-46.
Amherst, Massachusetts, June 15-17, 2023

eign colonization led to the development of Maltese as a distinct language shaped by Sicilian, Italian, and English influence. Written records from as early as 1240 acknowledge Maltese as its own language (Brincat, 2017), but it was not until 1934 that Maltese was declared an official language of Malta, along with English and Italian (Fabri, 2010). Italian was revoked as an official language in 1936, but its influence on the Maltese lexicon and grammar remains.

Much of the existing literature on Maltese describes the language as having a "split lexicon" or a "hybrid morphology" (e.g., Spagnol, 2011; Borg and Gatt, 2017). These characterizations reflect an etymological divide in the lexical stock. Semitic nouns in the language mostly form the plural with Semitic affixes or root-and-pattern templates, while non-Semitic nouns show a less strong tendency to form the plural with non-Semitic affixes. At the same time, hundreds of non-Semitic nouns inflect using Semitic patterns and are found in nearly all plural classes (Borg and Azzopardi-Alexander, 1997). Integration in the opposite direction is also found for a smaller number Semitic nouns which inflect using non-Semitic affixes. Maltese thus represents a partial, but not total, example of what has variously been called a "stratal effect" (Gardani, 2021) or "code compartmentalization" (Friedman, 2013) or "compartmentalized morphology" (Matras, 2015), in which native and borrowed morphological exponents in a language are restricted to applying to lexemes of the same etymological origin.

It is common in contact linguistics to describe outcomes of language contact as compositions of distinct linguistic systems, even in cases of extensive borrowing or codeswitching (e.g., Myers-Scotton, 1997; Gardani, 2020). Such descriptions are sometimes intended as theoretical analyses. For example, Gardani (2021) treats the stratal effect not simply as an empirically observable pattern, but as a synchronic constraint within the grammar that is psychologically real for speakers: "... a restriction on the application domain of non-native morphological formatives in a recipient language..." (Gardani, 2021, 132) that enforces the boundaries of etymologically-defined morphological subsystems.

However, we find the a priori assumption that stratal effects reflect distinct and psychologically real morphological subsystems to be problematic inasmuch as it conflates the property to be explained – that language contact can result (to greater or lesser degree) in compartmentalized morphology – with the mechanisms that produce and reinforce that compartmentalization. Stated differently, reification of the stratal effect as a mechanism of the grammar obscures important questions: Given that speakers do not generally know the etymological origins of words, how do they classify words into morphological patterns? What is the relationship between the processes that they use to do this and the stratal effect (or lack thereof) as an empirically observable outcome of language contact?

In this study we examine the (partial) stratal effect found in Maltese noun morphology, examining its relationship to factors known to be important outside of contact situations to how speakers classify words into morphological patterns. In particular we analyze the relative strength of a word's phonology and etymology as predictors of its nominal plural morphology and look at the relevance of these factors for the organization of the Maltese lexicon. It is important to note that we are not interested in etymology directly and we do not assume that speakers have or use direct knowledge of the etymology of words. We instead use etymology as a way to estimate the influence of conservative forces on morphological classification. We assume that the predictive power of etymology applies to words which have retained their etymological plurals, in some cases resisting pressures to conform to other parts of the language system. The conservative forces which resist these pressures include token frequency (Krause-Lerche, 2022).

Additionally, as a related question, we ask whether there is evidence in Maltese for distinct morphological subsystems ("hybrid morphology") in theoretical terms. This question is interesting in part because characterizations of Maltese as having hybrid morphology have also suggested, sometimes explicitly, that the non-concatenative morphology native to Semitic languages should be analyzed as distinct from concatenative morphology, both Semitic and non-Semitic. Moreover, research on morphological integration in Semitic languages has tended to focus specifically on the extent to which foreign words make use of native root-and-template morphology, as compared to affixation (e.g., Bensoukas, 2018; Ziani, 2020). However, since the vast majority of suffixal allomorphs in Maltese are of Semitic origin, division of the lexicon along etymo-

logical lines does not correspond to a split according to concatenative vs. non-concatenative morphology, as is sometimes implied. We test whether morphological type is a distinct factor in the stratal effect. Specifically, we ask whether there is support for analyzing root-and-pattern (templatic) plural morphology and affixal plural morphology as distinct subsystems.

We compare the results of two models: the first uses a lexeme's phonology and etymology to predict its concatenative type, either affixal or templatic. The second uses the same information to predict its inflectional allomorph, i.e., the specific affix or template found on the lexeme's plural form. Comparisons across factors within each model provide insight into the extent to which phonology and etymology are informative about plural morphology, and thus are likely to have played a role in the development of the language over centuries of contact with speakers of non-Semitic languages. Comparisons across the two models offer insight into the extent to which templatic and affixal morphological patterns operate as distinct subsystems in Maltese.

3 Analogy and Language Change

We take an analogical approach, using the term *analogy* to refer broadly to any similarity-based, paradigmatic influence of one word on the morphological behavior of another. The importance of analogy as a mechanism of language change is well established in the field of historical linguistics (Anttila, 1977; Hock, 1991; Fertig, 2013; Joseph, 2013), but it is most often discussed with respect to its role in language-internal change, independent of the effects of language contact. In contact linguistics, the idea that (phonologically-based) analogy plays a role in whether and how borrowed words are morphologically integrated into a recipient language has a long history, going back to at least Haugen (1950) and Weinreich (1953). However, most analyses of lexical and morphological borrowing focus on the potential and observed outcomes of contact (see Matras and Adamou, 2020, for an overview), often with little to no discussion of the exact ways in which analogy is hypothesized to play a role.

To examine the role of analogy, we take a cue from Matras (2009), who proposes a usage-based model of language contact in which a multilingual individual draws on a unified repertoire of linguistic resources. In this section we elaborate on how such a perspective can help in understanding the role of analogy, specifically analogical classification, in contact-induced morphological change and the development of the Maltese lexicon.

3.1 The Paradigm Cell Filling Problem

Analysis of the analogical mechanism hypothesized to drive morphological integration in contact may be understood as an extension of the Paradigm Cell Filling Problem (PCFP), a line of research in theoretical morphology that seeks to identify the information available to speakers that allows them to infer and produce grammatically inflected surface forms (Ackerman et al., 2009). Most quantitative analyses of the PCFP to date take an analogical approach: speakers are hypothesized to rely on emergent similarities and paradigmatic relations among previously-acquired words in the lexicon to inform their intuitions when inflecting or processing rare or novel word forms (see, e.g., Ackerman et al., 2009; Sims and Parker, 2016; Guzmán Naranjo, 2020; Parker et al., 2022).

Matras's (2009) usage-based model of language contact is directly compatible with analogical approaches to the PCFP. Since multilingual speakers are assumed to have access to a unified linguistic repertoire corresponding to all of their languages, this full repertoire may be drawn upon to make morphological generalizations. Combinations of generalizations from different languages during speech production may result in linguistic innovations or morphologically adapted "nonce borrowings" (Poplack et al., 1988). Over time, some of these may be conventionalized and perpetuated throughout the larger speech community, leading to contact-induced language change.

We may therefore specify the PCFP with respect to language contact as follows: what guides speakers' grammatical intuitions when adapting and integrating lexemes in multilingual contexts, and how may conventionalized integration of borrowed linguistic material affect the intuitions of a monolingual speaker when producing inflected word forms?

3.2 Computational Modeling of the PCFP

A number of recent studies in computational linguistics have applied machine learning methods to analyze the kinds and amounts of information that may be available to speakers when solving the PCFP (in monolingual contexts). For example, Guzmán Naranjo (2020) uses a Long Short-

term Memory Network (LSTM, Hochreiter and Schmidhuber, 1996) to quantify the respective informativity of stem phonology, lexical semantics, and affixal exponents as predictors of nominal inflection class organization in Russian. His results indicate that while each factor contributes predictive information, more information about inflection class is contributed by stem phonology than by any individual affix. Furthermore, the contributions of the three predictors are additive, indicating a level of nonredundancy in their informativity.

Williams et al. (2020) also employ the representational power of an LSTM to quantify the extent to which phonology and lexical semantics are predictive of a noun's declension class in German and Czech. As opposed to model accuracy, they measure the amount of Mutual Information, in bits, shared by phonology, semantics, and declension class systems in each language. They find that, while phonology is more predictive than semantics overall in both languages, the relative informativity of phonology and semantics varies greatly across the two languages and across individual declension classes within each language.

Dawdy-Hesterberg and Pierrehumbert (2014) take an analogical approach to modeling plural formation in Modern Standard Arabic. The authors use a Generalized Context Model (GCM, Nosofsky, 1990) to quantify the extent to which phonological factors, specifically similarities in consonant-vowel (CV) template (a.k.a. "broken plural" allomorph), segmental properties (in terms of natural classes), and lexical gang size (Alegre and Gordon, 1999), predict the form of a plural noun in Arabic. Their results indicate that all three factors are predictive to varying degrees, suggesting phonological representations that are both fine-grained, i.e., at the segmental level, and coarse-grained, i.e., with respect to gang size and CV template, may serve as a basis for analogical processing and morphological organization in Arabic.

Finally, Nieder et al. (2021a,d) use both computational and psycholinguistic methods to investigate the role of analogical classification in the nominal plural system of Maltese. The authors find that plural forms in Maltese may be predicted with a reasonable degree of accuracy based on their phonological similarity to attested plural forms, modulated by the frequency distribution of plural allomorphs in the language. However, the authors do not specifically measure etymology as a predictor, leaving open the question of how non-Semitic words were integrated into the morphological system. In other words, it is unclear from their results whether phonology is predictive independently of etymology, or only as an indicator of etymological origin.

4 Methods

The current study adapts the methods proposed by Williams et al. (2020) to quantify the relative contributions of phonology and etymology as predictors of inflectional organization in Maltese. We use a character-level LSTM classifier trained to make inferences about a word's plural class by abstracting over the phonology of each word form as a whole. We then quantify the influence of phonology on Maltese nominal plural inflection using Mutual Information, an information theoretic measure of interpredictability among two or more systems. We compare our results to the predictive strength of the word's etymological origin using the same measures, quantifying the balance of analogical and conservative factors hypothesized to shape the integration of foreign lexemes into the grammar.

4.1 Data

This study merges data from two collections compiled by Nieder et al. (2021b,c) into a single dataset consisting of 3,174 singular-plural noun pairs. Each pair is tagged for etymological origin, either Semitic or non-Semitic. The original data was manually compiled from the MLRS Korpus Malti v. 2.0 and 3.0 (Gatt and Čéplö, 2013) and supplemented with Schembri's (2012) collection of Maltese CV templates. Etymological information was sourced from a digitized version of Aquilina's (2006) Maltese-English dictionary. Plural nouns in the data are classified as taking one of 12 different suffixes ("sound plurals") or 11 different non-concatenative CV templates ("broken plurals"), forming a nominal plural inflection system composed of 23 different inflection classes (Nieder et al., 2021b). Maltese is the only standardized Semitic language written in a Latin script, using an orthography that "represents the phonology of the language admirably" according to Hoberman (2007, 258). For this reason, we analyze nouns using their original orthography, as in Williams et al. (2020).

Over 135 nouns in the dataset take more than one plural form. Of these, 78 nouns may take both

	Non-Semitic Lexeme	Semitic Lexeme	Total (%)
Non-Semitic Affix	1,274	21	42%
Semitic Affix	416	684	35%
Semitic Template	240	537	23%
Total (%)	62%	38%	100%

Table 1: Distribution of Maltese nominal plural allomorphs by lexeme etymology and concatenative type

broken and sound plurals. In this study, we account for these nouns by representing each pair separately at the allomorph level, whereas in the binary prediction model of the lexeme's concatenative type (concatenative vs. non-concatenative) we include a noun only once per type. For example, the word LIBSA 'dress' may take the sound plural *libsiet* and the broken plurals *lbies* and *lbiesi*. The lexeme LIBSA is therefore included in the model three times in the allomorph prediction setting, but only twice in the type prediction setting.

Following Williams et al. (2020), we remove all classes with fewer than 20 lexemes, leaving a total of 13 plural allomorph classes in our model. Table 1 shows the full distribution of allomorphs according to etymology and concatenative type. Note that lexemes that take more than one allomorph are counted more than once.

4.2 Formal Notation

Following Williams et al. (2020), we can define a lexeme as a tuple (w_i, e_i, c_i) where for the i^{th} lexeme, w_i = the lexeme's phonological form, e_i = the lexeme's etymological origin, and c_i = the lexeme's inflection class. We assume the lexemes follow a probability distribution $p(w, e, c)$, approximated by the corpus. We can define the space of K inflection classes as $\mathcal{C} = \{1, ..., K\}$, so that $c_i \in \mathcal{C}$ and define C as the random variable associated with $\mathcal{C}$. For a set of lexemes derived from N etymological origins, we can define an etymological space as $\mathcal{E} = \{1, ..., N\}$ so that $e_i \in \mathcal{E}$ and define E as the random variable associated with $\mathcal{E}$. Each noun may be associated with one of two genders g_i from the space of genders $\mathcal{G}$ specific to Maltese. Finally, we define the space of word forms as the Kleene closure over a language's alphabet Σ, so that $w_i \in \Sigma^*$, with W as the random variable associated with Σ^*.

4.3 Mutual Information (MI)

Mutual Information (MI) is an information theoretic measure that quantifies the degree of interpredictability among two or more systems. For example, the MI shared by the nominal plural inflection class system C and phonological system W in Maltese may be calculated as follows:

$$\mathrm{MI}(C; W) = \mathrm{H}(C) - \mathrm{H}(C|W) \quad (1)$$

This may be generalized to consider the amount of redundant information shared by inflection class, phonology, and etymology E as follows:

$$\mathrm{MI}(C; E; W) = \mathrm{MI}(C; W) - \mathrm{MI}(C; W|E) \quad (2)$$

Because a language's grammatical gender system is known to interact with its inflectional morphology in non-deterministic ways (Corbett and Fraser, 2000), we follow Williams et al. (2020) and condition all relevant measures on gender:

$$\mathrm{MI}(C; W|G) = \mathrm{H}(C|G) - \mathrm{H}(C|W, G) \quad (3)$$

The intuitive reasoning behind Equations 1 - 3 may be seen in Figure 1, in which each colored circle represents H|G, the total entropy, conditioned on gender, of the three interacting systems under analysis.

Finally, since our corpus is only a sample of the language, we note that all calculations are estimates. However, while estimates over the finite inflection class and etymology systems can be empirically calculated using the corpus, the infinite number of possible word forms in the Σ^* means calculations involving W must be further approximated. Methods for estimating the entropy of both kinds of systems are described in detail in the following sections.

4.4 Techniques for Estimating Entropy

We use plug-in estimation to obtain entropy values for C and E, calculating the distribution $p(c)$ for $c \in C$ (or alternatively, $p(e)$ for $e \in E$) and using this to estimate $\mathrm{H}(C)$ in Equation 1 above.

4.5 Approximating Conditional Entropy

$\mathrm{H}(C|E)$ may be similarly calculated using plug-in estimation. However, given the infinite number of possible word forms in Σ^*, an estimate for $\mathrm{H}(C|W)$ cannot be calculated directly from the

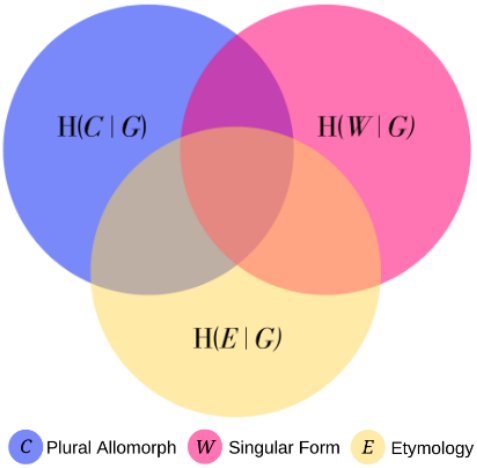


Figure 1: Tripartite Mutual Information

corpus. We therefore approximate this value using cross-entropy, which has been mathematically proven to be an upper bound on conditional entropy (Brown et al., 1992). We use the cross-entropy loss obtained from a computational model that has been trained to predict the plural class c_i associated with a singular noun w_i to approximate the cross-entropy of the system:

$$\mathrm{H}(C|W) \leq -\frac{1}{M}\sum_{i=1}^{M}\log q(c_i|w_i) \qquad (4)$$

We note that as the amount of data in the corpus increases, i.e., as $M \to \infty$, the above value approaches the true cross-entropy value.

4.6 Normalized Mutual Information (NMI)

To compare results across models and across languages, we normalize MI values by dividing by the total entropy of the inflection class system. For example, the NMI shared by a Maltese noun's phonology and plural inflection may be calculated as:

$$\mathrm{NMI}(C;W) = \frac{\mathrm{MI}(C;W)}{\mathrm{H}(C)} \qquad (5)$$

4.7 Model Details

We adapt the LSTM classifier implemented in Williams et al. (2020) to estimate the probability that a plural class c is associated with a given input noun w of gender g, i.e., $q(c|w,g)$ in Equation 4. The model learns a set of character embeddings to represent the phonological forms of singular nouns as part of the training process. Gender is separately embedded and input into the model's initial hidden state. The model is trained using Adam (Kingma and Ba, 2015) with model hyperparameters, including the number of training epochs and the number and sizes of hidden layers, optimized using the Bayesian optimization technique implemented in Williams et al. (2020). The model then learns a probability distribution that serves to approximate $q(c|w,g)$.

Following training, we test the model on a held-out dataset and use the model's cross-entropy loss to serve as an approximate upper bound on the conditional entropy $\mathrm{H}(C|W,G)$. We use 10-fold cross validation to make full use of the dataset for our approximations. To estimate $q(c|w,e,g)$, we concatenate a binary character representing the word's etymology onto the end of the noun to serve as model input and follow the same procedure.

		TYPE	ALLO.
$\mathrm{H}(C\|G)$		0.81	2.65
$\mathrm{NMI}(C;W\|G)$		0.21	0.42
$\mathrm{NMI}(C;E\|G)$		0.13	0.22
$\mathrm{NMI}(C;E;W\|G)$		0.06	0.15
$\mathrm{NMI}(E;W\|G)$		0.61	0.61

Table 2: Normalized Mutual Information measures for plural class C defined with respect to TYPE vs. ALLOMORPH. NMI values involving C are normalized with respect to $\mathrm{H}(C|G)$, while $\mathrm{NMI}(E;W|G)$ is normalized with respect to $\mathrm{H}(E|G)$.

5 Results

NMI and $\mathrm{H}(C|G)$ values for C defined as concatenative type and plural allomorph, respectively, are presented in Table 2. The largest NMI value we obtain, $\mathrm{NMI}(E;W|G)$, indicates that more than half of the information needed to predict a word's etymology is shared with its phonology. In other words, it is often not difficult to guess the origin of a Maltese word based on how it sounds. Note that this value is consistent across models, as it does not depend on C.

5.1 Concatenative Type

Results for the model predicting a noun's concatenative type are in Table 2. Note first that the entropy $\mathrm{H}(C|G)$ of the plural inflection class system defined at the level of concatenative type is calculated to be 0.81, indicating that, given its gender, predicting whether a random Maltese noun takes concatenative or non-concatenative morphology is more

predictable than chance, although not by much. We find phonology, indicated by $\mathrm{NMI}(C;W|G)$, to be more predictive than etymology, indicated by $\mathrm{NMI}(C;E|G)$. Crucially, each of these bipartite NMI values exceeds the tripartite mutual information $\mathrm{NMI}(C;E;W|G)$ shared across all three systems. This indicates that while a non-trivial amount of predictive information is shared across all three systems, phonology and etymology are each predictive of concatenative type in partially non-redundant ways. This suggests that both analogical and conservative forces are likely to have played a role in the development of the Maltese nominal plural system.

5.2 Plural Allomorph

In an analogical model of inflection in which singular inflected forms and their plural counterparts share a direct relationship in the lexicon, the predictive principles structuring the morphological system are expected to be most evident when defining an inflection class system at the level of the allomorph.

We first note that the entropy $\mathrm{H}(C|G)$ calculated over the plural class distribution defined at the allomorph level is nearly three times as high as the entropy of C when defined as a noun's concatenative type. This is reflective of the higher degree of unpredictability associated with a non-uniform distribution of nouns over a greater number of inflection classes. When comparing across the allomorph and concatenative type models it is thus important to normalize for the fact that predicting allomorphs is more difficult than predicting concatenative type. However, even calculations normalized in this way show that the interpredictability among phonology, etymology, and plural inflection, indicated by the NMI values in Table 2, are all twice as high at the allomorph level as they are for concatenative type. In other words, a noun's singular form reduces the relative uncertainty about its plural allomorph twice as much as it reduces the uncertainty about whether that allomorph is concatenative. This suggests the analogical and conservative pressures hypothesized to shape morphological organization are more sensitive to correspondences at the word level than to typological similarities with respect to concatenativity.

Additionally, the general tendency found at the level of concatenative type still follows when classes are defined at the level of individual allomorphs: phonology shares more information with inflection class than does etymology, with each factor contributing some amount of non-redundant information. This illustrates one key advantage of the methods employed in this study, namely the ability to disentangle the independent contributions of either predictor from the degree to which both exert redundant organizational pressure towards the same end.

For example, given the fact that phonology and etymology are themselves mutually informative, we cannot uniquely interpret either bipartite measure of MI, that is, $\mathrm{NMI}(C;W|G)$ or $\mathrm{NMI}(C;E|G)$, as indicative of the forces hypothesized to shape the integration of linguistic material in contact. Rather, evidence for analogical structuring of the Maltese plural system at the allomorph level is specifically indicated by the positive difference between $\mathrm{NMI}(C;W|G)$ and $\mathrm{NMI}(C;E;W|G)$. Conservative pressures, such as those associated with high token-frequency items (Krause-Lerche, 2022), are similarly indicated by the extent to which $\mathrm{NMI}(C;E|G)$ exceeds $\mathrm{NMI}(C;E;W|G)$.

5.3 Variation Across Allomorph Classes

Closer examination of the model's predictions reveals an effect of type frequency, with larger inflection classes predicted more often than smaller classes. Table 3 reports the accuracy of all models in which singular noun phonology W is a predictor. Since all models achieve an overall accuracy above a majority baseline, the NMI values we obtain may be reliably interpreted as empirical minimums. However, as can be seen in Figure 2, the model's incorrect predictions do not clearly distinguish between sound and broken classes; nouns with a sound plural allomorph may be misclassified as taking a broken plural template, and nouns taking a broken plural may be incorrectly predicted to take a sound plural.

If speakers are sensitive to differences between concatenative and non-concatenative allomorphs grouped into high-level macro classes (morphological subsystems), we might expect some degree of observable within-class coherence with respect to either or both of the phonology and etymology of words exhibiting a particular morphological behavior. Specifically, we would expect a pattern of predictions in which the LSTM is able to first identify a lexeme's concatenative type before pre-

Target	Model	Accuracy
ETYM. (E)	MI($E;W|G$)	0.90
	Baseline	0.62
TYPE (C)	MI($C;W|G$)	0.80
	MI($C;E;W|G$)	0.81
	Baseline	0.77
ALLOMORPH (C)	MI($C;W|G$)	0.65
	MI($C;E;W|G$)	0.68
	Baseline	0.40

Table 3: Model accuracy for all models predicting Etymology E or Plural Class C (Type vs. Allomorph) using the Phonology W of singular nouns in Maltese

dicting, possibly incorrectly, an allomorph of that specific type.

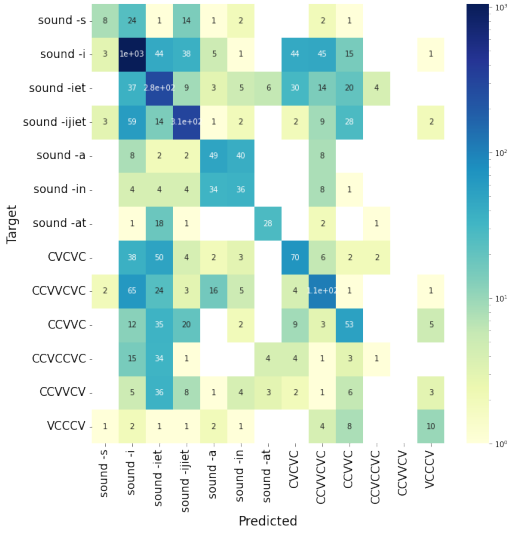


Figure 2: Confusion matrix: predicting plural allomorph from singular phonology and gender

Instead, as seen in Figure 2, we do not find such evidence. Rather, we find evidence for coherence at the allomorph level, specifically, for phonological patterns as a predictor of inflectional organization and driver of inflectional behavior at the allomorph level.

Finally, as in Williams et al. (2020), we also conduct an analysis of the partial Pointwise Mutual Information (PMI) shared between phonology W and class C with respect to the surprisal $\mathrm{H}(C = c|G)$ for each class, defined at the allomorph level. Figure 3 shows this distribution, with allomorph classes presented in order of increasing type frequency (and thus decreasing surprisal). We note that Maltese noun classes are each only partially predictable given the phonology of words

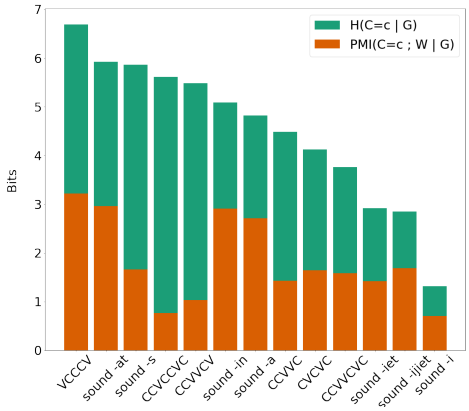


Figure 3: Partial Pointwise Mutual Information (PMI) shared by word form and class for each allomorph class

belonging to them, regardless of class size or etymological origin.

6 Discussion

In this paper we used an LSTM to help estimate the kinds and amounts of information that may be available to speakers when "solving" the PCFP. Overall, our results provide quantitative evidence for the role of both word phonology and etymology (as a stand-in for conservative factors) in shaping the Maltese lexicon.

Specifically, we found that the extent to which a Maltese singular noun's phonology predicts its plural morphology exceeds that of etymology in non-redundant ways. This suggests that analogical pressures from phonological correspondences across the lexicon shape nominal plural inflection in Maltese, independently of the etymological source language for some word or morphological pattern.

Our results also show an independent contribution of etymology as a predictor. We hypothesize that this captures conservative pressures theorized to resist analogical change, including token frequency (Krause-Lerche, 2022). It may also reflect associative correlations from the use of lexemes of a common etymology in similar contexts, strengthening their coherence as a subsystem in the multilingual repertoire and encouraging the maintenance of a noun's original morphology. Further work is needed to investigate these possibilities.

In language contact situations such as that of Maltese, it is likely that an influx of foreign lexemes and increased productivity of foreign affixes affect both the size and character (e.g., phonology)

of nominal plural classes relative to each other over time. This in turn is likely to affect subsequent classification and integration of words into the inflectional morphology of the language.

In general, our results do not support characterizations of Maltese in which concatenative and non-concatenative morphologies co-exist as discrete systems within the lexicon. While a singular noun's phonology and etymology are each somewhat predictive of its concatenative type, they are twice as predictive of the actual plural allomorph(s) with which the lexeme is associated. This suggests that systematic relationships at the word level organize the morphology of Maltese, in turn shaping the language as new words are integrated and inflected.

7 Conclusion

This study extends previous work in information theory, computational modeling, and theoretical morphology to provide quantitative evidence for the role of phonology as an analogical force in the morphological organization of Maltese. We ground this in a usage-based account of multilingualism and contact-induced change in which speakers are hypothesized to make use of analogical reasoning, among other language-general cognitive functions, when integrating novel words and patterns within a unified linguistic repertoire. The same processes that guide synchronic language use are proposed to be responsible for the diachronic effects of contact-induced language change. Specifically, it is hypothesized that speakers draw on similarities across multiple dimensions – including but not limited to phonological patterns, semantic and indexical meaning, pragmatic function, and contexts of use – to collaboratively construct and adapt grammatical systems of linguistic communication over time.

In the case of Maltese, our findings indicate that while a lexeme's phonology and etymology are themselves highly interpredictable, each contributes non-redundant information to reduce uncertainty when predicting the lexeme's plural inflection. While the etymology of a noun is somewhat predictive of its plural inflection, the word's phonology plays a much greater role. This synchronic analysis has diachronic implications. Our results suggest that analogical pressures from phonological similarities across the lexicon may have guided speakers' inflectional behavior when code mixing over the course of the development of the language to result in the conventionalized forms observed in modern Maltese. However, further diachronic study is needed to confirm this interpretation.

Contrary to a hypothesis in which concatenative and non-concatenative systems operate as separate subsystems within a "split" or "hybrid" morphology, our results indicate correspondences at the level of individual wordforms and affixes are driving speakers' morphological behavior. Specifically, the phonology and etymology of a lexeme are twice as predictive of its plural allomorph than its concatenative type. Further investigation into Maltese nouns attested to take plural forms of both concatenative types may provide additional insight into the ways in which concatenative type affects speakers' behavior, if at all. Future work should also consider additional factors known to shape inflection class systems, for example by integrating semantic word vectors into the model. Finally, additional comparisons implementing these methods across corpora in a variety of languages will continue to shed light on the factors shaping morphological systems cross-linguistically.

Acknowledgments

We thank Jessica Nieder and Adam Ussishkin for generously sharing an abundance of digital resources and helpful feedback, Sarah Caruana for her insight into the Maltese language, and Christian Clark and Andrew Duffy for their contributions to an initial version of this project.

This material is based on work supported by the National Science Foundation under grant BCS-2217554 (*Neural discovery of abstract inflectional structure*, PI Micha Elsner, Co-PI Andrea Sims).

References

Farrell Ackerman, James P. Blevins, and Robert Malouf. 2009. Parts and wholes: Implicative patterns in inflectional paradigms. In James P. Blevins and Juliette Blevins, editors, *Analogy in grammar: Form and acquisition*, pages 54–82. Oxford University Press, Oxford.

Maria Alegre and Peter Gordon. 1999. Rule-based versus associative processes in derivational morphology. *Brain and Language*, 68(1-2):347–354.

Raimo Anttila. 1977. *Analogy*. The Hague: Mouton.

Joseph Aquilina. 2006. *Concise Maltese-English, English-Maltese dictionary*. Midsea Books, Sta Venera, Malta.

Karim Bensoukas. 2018. Concurrent cognate and contact-induced plural traits in Afro-Asiatic:

Amazigh *id-* and Arabic *-at* plurals. *International Journal of Arabic Linguistics*, 4(1):59–102.

Albert J. Borg and Marie Azzopardi-Alexander. 1997. *Maltese*. Routledge, London.

Claudia Borg and Albert Gatt. 2017. Morphological analysis for the Maltese language: The challenges of a hybrid system. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 25–34, Valencia, Spain. Association for Computational Linguistics.

Joseph M. Brincat. 2011. *Maltese and other languages: A linguistic history of Malta*. Midsea Books, Sta Venera, Malta.

Joseph M. Brincat. 2017. Maltese: Blending Semitic, Romance and Germanic lexemes. *Lexicographica*, 33(2017):207–224.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Jennifer C. Lai, and Robert L. Mercer. 1992. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40.

Greville G. Corbett and Norman M. Fraser. 2000. Gender assignment: A typology and a model. In Gunter Senft, editor, *Systems of nominal classification*, pages 293–325. Cambridge University Press, Cambridge.

Lisa Garnand Dawdy-Hesterberg and Janet Breckenridge Pierrehumbert. 2014. Learnability and generalisation of Arabic broken plural nouns. *Language, Cognition and Neuroscience*, 29(10):1268–1282.

Ray Fabri. 2010. Maltese. *Revue belge de Philologie et d'Histoire*, 88(3):791–816.

David L. Fertig. 2013. *Analogy and morphological change*. Edinburgh University Press, Edinburgh.

Victor A. Friedman. 2013. Compartmentalized grammar: The variable (non)-integration of Turkish verbal conjugation in Romani dialects. *Romani Studies*, 23:107–120.

Francesco Gardani. 2020. Borrowing matter and pattern in morphology: An overview. *Morphology*, 30(4):263–282.

Francesco Gardani. 2021. On how morphology spreads. *Word Structure*, 14(2):129–147.

Albert Gatt and Slavomír Čéplö. 2013. Digital corpora and other electronic resources for Maltese. In *Proceedings of the International Conference on Corpus Linguistics*, pages 96–97, Lancaster, UK. UCREL Lancaster.

Matías Guzmán Naranjo. 2020. Analogy, complexity and predictability in the Russian nominal inflection system. *Morphology*, 30(3):219–262.

Einar Haugen. 1950. The analysis of linguistic borrowing. *Language*, 26(2):210–231.

Robert D. Hoberman. 2007. Maltese morphology. *Morphologies of Asia and Africa*, 1:257–281.

Sepp Hochreiter and Jürgen Schmidhuber. 1996. LSTM can solve hard long time lag problems. *Advances in Neural Information Processing Systems*, 9:473–479.

Hans Henrich Hock. 1991. *Principles of historical linguistics*. De Gruyter Mouton, Berlin.

Brian D. Joseph. 2013. On phonically based analogy. *Working papers in linguistics*, 60:11–20.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR2015)*, San Diego, CA.

Anne Krause-Lerche. 2022. Conservation in ongoing analogical change: The measurement and effect(s) of token frequency. *Corpus Linguistics and Linguistic Theory*, 18(1):77–114.

Yaron Matras. 2009. *Language contact*. Cambridge University Press, Cambridge.

Yaron Matras. 2015. Why is the borrowing of inflectional morphology dispreferred? In Francesco Gardani, Peter Arkadiev, and Nino Amiridze, editors, *Borrowed morphology*, pages 47–80. De Gruyter Mouton, Berlin.

Yaron Matras and Evangelia Adamou. 2020. Borrowing. In Evangelia Adamou and Yaron Matras, editors, *The Routledge handbook of language contact*, pages 237–251. Routledge, London.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Clarendon, Oxford.

Jessica Nieder, Fabian Tomaschek, Enum Cohrs, and Ruben van de Vijver. 2021a. Modelling Maltese noun plural classes without morphemes. *Language, Cognition and Neuroscience*, 37(3):381–402.

Jessica Nieder, Fabian Tomaschek, Enum Cohrs, and Ruben van de Vijver. 2021b. Modeling Maltese noun plural classes without morphemes: Supplemental materials. Available online: https://osf.io/pyf7b/.

Jessica Nieder, Ruben van de Vijver, Yu-Ying Chuang, and R. H. Baayen. 2021c. A discriminative lexicon approach to word comprehension, production and processing: Maltese plurals: Supplemental materials. Available online: https://osf.io/rxsbu/.

Jessica Nieder, Ruben van de Vijver, and Holger Mitterer. 2021d. Knowledge of Maltese singular–plural mappings. *Morphology*, 31(2):147–170.

Robert M. Nosofsky. 1990. Relations between exemplar-similarity and likelihood models of classification. *Journal of Mathematical Psychology*, 34(4):393–418.

Jeff Parker, Robert Reynolds, and Andrea D. Sims. 2022. Network structure and inflection class predictability: Modeling the emergence of Marginal Detraction. In Andrea D. Sims, Adam Ussishkin, Jeff Parker, and Samantha Wray, editors, *Morphological diversity and linguistic cognition*, pages 247–281. Cambridge University Press, Cambridge.

Shana Poplack, David Sankoff, and Christophe Miller. 1988. The social correlates and linguistic processes of lexical borrowing and assimilation. *Linguistics*, 26(1):47–104.

Tamara Schembri. 2012. *The broken plural in Maltese: A description. Il-lingwa tagħna.* Brockmeyer, Bochum.

Andrea D. Sims and Jeff Parker. 2016. How inflection class systems work: On the informativity of implicative structure. *Word Structure*, 9(2):215–239.

Michael Spagnol. 2011. *A tale of two morphologies: Verb structure and argument alternations in Maltese*. Ph.D. thesis, University of Konstanz.

Uriel Weinreich. 1953. *Languages in contact: Findings and problems*. De Gruyter Mouton, Berlin.

Adina Williams, Tiago Pimentel, Hagen Blix, Arya D. McCarthy, Eleanor Chodroff, and Ryan Cotterell. 2020. Predicting declension class from form and meaning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6682–6695, Online. Association for Computational Linguistics.

Zoubida Ziani. 2020. The morphology of borrowings and its relevance to lexical organization in Moroccan Arabic. *International Journal of Arabic Linguistics*, 6(1-2):242–268.

A Nominal Plural Allomorphs in Maltese

Sound Plural

Singular	Plural	Gloss	Allomorph
karta	*karti*	'paper'	*-i*
omm	*ommijiet*	'mother'	*-ijiet*
rixa	*rixiet*	'feather'	*-iet*
giddieb	*giddieba*	'liar'	*-a*
meħlus	*meħlusin*	'freed'	*-in*
kuxin	*kuxins*	'cushions'	*-s*
triq	*triqat*	'street'	*-at*
sid	*sidien*	'owner'	*-ien*
baħri	*baħrin*	'sailor'	*-n*
ħati	*ħatjin*	'guilty'	*-jin*
spalla	*spallejn*	'shoulder'	*-ejn*
sieq	*saqajn*	'foot'	*-ajn*
qiegħ	*qiegħan*	'bottom'	*-an*

Table 4: Sound plural allomorphs in Maltese, from Nieder et al. (2021b)

Broken Plural

Singular	Plural	Gloss	Allomorph
fardal	*fradal*	'apron'	CCVVCVC
birra	*birer*	'beer'	(C)CVCVC
kbir	*kbar*	'big'	CCVVC
ftira	*ftajjar*	'type of bread'	CCVjjVC
bitħa	*btieħi*	'yard'	CCVVCV
sider	*isdra*	'chest'	VCCCV
marid	*morda*	'sick person'	CVCCV
għodda	*għodod*	'tool'	(għ)VCVC
elf	*eluf*	'thousand'	VCVC
għaref	*għorrief*	'wise man'	CVCCVVC(V)
għama	*għomja*	'blind person'	(għ)VCCV

Table 5: Broken plural allomorphs in Maltese, from Nieder et al. (2021b)

CANDS: A Computational Implementation of Collins and Stabler (2016)

Satoru Ozaki
University of Massachusetts Amherst
sozaki@umass.edu

Yohei Oseki
University of Tokyo
oseki@g.ecc.u-tokyo.ac.jp

Abstract

Syntacticians must keep track of the empirical coverages and the inner workings of syntactic theories, a task especially demanding for minimalist syntacticians to perform manually and mentally. We believe that the computational implementation of syntactic theories is desirable in that it not only (a) facilitates the evaluation of their empirical coverages, but also (b) forces syntacticians to specify their inner workings. In this paper, we present CANDS, a computational implementation of **C**ollins **AND** **S**tabler (2016) in the programming language Rust. Specifically, CANDS consists of one main library, `cands`, as well as two wrapper programs for `cands`, `derivck` and `derivexp`. The main library, `cands`, implements key definitions of fundamental concepts in minimalist syntax from Collins and Stabler (2016), which can be employed to evaluate and extend specific syntactic theories. The wrapper programs, `derivck` and `derivexp`, allow syntacticians to check and explore syntactic derivations through an accessible interface.[1]

1 Introduction

Syntax typically involves developing a new theory or revising an existing theory in order to explain certain data. A syntactician needs to be able to compare the theories in terms of their empirical coverage and understand all the details of these theories. These are challenging prerequisites to attain for minimalist syntacticians (Chomsky, 1995). This is partly due to the lack of consensus on the exact mechanism of minimalist syntactic theory, despite many efforts to formalize it (e.g., Veenstra 1998; Kracht 1999, 2001, 2008; Frampton 2004; Collins and Stabler 2016), and partly due to the constant source of subtle revisions to this theory.

We believe that the computational implementation of syntactic theories would help minimalist syntacticians understand their empirical coverages and inner workings. This idea has been explored in the LFG and HPSG literature with their rich histories of grammar engineering (e.g., Bierwisch 1963; Zwicky et al. 1965; Müller 1999; Butt 1999; Bender et al. 2002, 2008, 2010; Fokkens 2014; Müller 2015; Zamaraeva 2021; Zamaraeva et al. 2022). In comparison, there is less effort on the computational implementation of syntactic theories in the minimalist literature, with some exceptions (e.g., Fong and Ginsburg, 2019). In this paper, we present CANDS (pronounced /kændz/), a computational implementation of **C**ollins **AND** **S**tabler (2016) (henceforth C&S) in the programming language Rust. The main library, `cands`, implements key definitions of fundamental concepts in minimalist syntax from Collins and Stabler (2016), which itself is a formalization of minimalist syntax. We hope that `cands` can be employed to evaluate and extend specific syntactic theories.

In addition, to make `cands` accessible to minimalist syntacticians who are not familiar with Rust, we also provide two wrapper programs for `cands` which allow syntacticians to check and explore syntactic derivations through an accessible interface: the derivation checker `derivck`, and the derivation explorer `derivexp`.

This paper is organized as follows. In Section 2, we review key definitions of fundamental concepts in minimalist syntax from C&S. In Section 3, we introduce the main library, `cands`, as well as two wrapper programs, `derivck` and `derivexp`, illustrating their usage with example codes and screenshots. In Section 4, we demonstrate how `cands` can be employed to evaluate syntactic theories with two particular formulations of the Subject Condition. In Section 5, we show how `cands` can be used to extend syntactic theories with two particular implementations of the syntactic operation Agree. We discuss future work in Section 6 and conclude the paper in Section 7.

[1] Our software is available at https://github.com/osekilab/CANDS.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 47-68.
Amherst, Massachusetts, June 15-17, 2023

2 Collins and Stabler (2016)

Collins and Stabler (2016) provide a precise formulation of minimalist syntax. In this section, we review some key definitions in their work.

Universal Grammar (UG) is a 6-tuple $\langle \text{PHON-F}, \text{SYN-F}, \text{SEM-F}, \text{Select}, \text{Merge}, \text{Transfer} \rangle$, where the first three elements specify the universal sets of phonological, syntactic and semantic features respectively, and the last three elements are syntactic operations.

An *I-language* is as a 2-tuple $\langle \text{Lex}, \text{UG} \rangle$ where Lex is a lexicon, i.e., a finite set of lexical items, and UG is some Universal Grammar.

A *lexical item* (LI) is a 3-tuple $\langle \text{SEM}, \text{SYN}, \text{PHON} \rangle$, where $\text{SEM} \subseteq \text{SEM-F}$, $\text{SYN} \subseteq \text{SYN-F}$ and $\text{PHON} \in \text{PHON-F}^*$. [2]

A *lexical item token* (LIT) is a 2-tuple $\langle \text{LI}, k \rangle$, where LI is a LI and k an *index*. This index is used to distinguish between multiple occurrences of the same LI related by movement.

Syntactic objects (SO) are inductively defined. A SO is one of three things: (a) a LIT, (b) the result of the syntactic operation $\text{Cyclic-Transfer}(\text{SO})$ for some syntactic object SO, or (c) a set of SOs.

A *lexical array* (LA) is a set of LITs, and a *workspace* W is a set of SOs. A *stage* is a 2-tuple $\langle \text{LA}, W \rangle$ of lexical array LA and workspace W.

The syntactic operations Select, Merge and Transfer are defined as functions. For example, for some stage $S = \langle \text{LA}, W \rangle$ and LIT $A \in \text{LA}$,

$$\text{Select}(A, S) = \langle \text{LA} \setminus \{A\}, W \cup \{A\} \rangle.$$

Cyclic-Transfer, which was used in the above definition of SOs, is a special unary case of Transfer, which is a binary operation.

The central definition in C&S is that of a derivation. A sequence of stages $S_1, \cdots, S_n$ with each $S_i = \langle \text{LA}_i, W_i \rangle$ is a *derivation* from lexicon L if (a) all LIs from the initial lexical array LA_1 come from L, (b) the initial workspace W_1 is empty, and (c) each subsequent stage S_{i+1} is derived from the previous stage S_i by an appropriate application of some syntactic operation. The conditions involved in (c) limit the generative capacity of the theory. For example, the conditions on Merge enforce that, if S_{i+1} is derived from S_i by $\text{Merge}(A, B)$, then $A \in W_i$, and either A contains B or $B \in W_i$. The first disjunct "A contains B" allows internal Merge, and the second disjunct "$B \in W_i$" allows external Merge. Certain patterns of Merge, such as sideward Merge, are disallowed in this formulation.

[2] PHON-F^* is the set of (potentially empty) sequences whose elements come from PHON-F, i.e., $\bigcup_{k=0}^{\infty} \text{PHON-F}^k$.

3 CANDS

CANDS consists of the main library, `cands`, and two wrapper programs for `cands`, `derivck` and `derivexp`. They are all developed in the programming language Rust.

3.1 `cands`

`cands` is a library that implements and exposes most concepts defined in C&S. We provide a full list of implemented definitions in Appendix A.

Figure 1 shows the Rust code that uses `cands` to create a SO. This SO is a LIT, with index 37 and a LI that consists of the semantic features {[M]}, the syntactic features {[D]}, and the phonological features ⟨[Mary]⟩. `SyntacticObject` is an enum type defined in `cands`, which comes in three variants: LITs, sets and results of Cyclic-Transfer. Here, we use `SyntacticObject::LexicalItemToken` to construct a LIT variant. `cands` also defines the struct types `LexicalItemToken`, `LexicalItem` and `Feature`, each of which is associated with a `new` function that constructs an object of each type. `Set` and `Vec` are container types defined in the Rust standard library, and their associated `from` functions create sets and vectors.[3]

```
SyntacticObject::LexicalItemToken(
  LexicalItemToken::new(
    LexicalItem::new(
      Set::from([Feature::new("M")]),
      Set::from([Feature::new("D")]),
      Vec::from([Feature::new("Mary")])
    ), 37
  )
)
```

Figure 1: Code to create a SO.

`cands` defines many macros, which help reduce boilerplate code. For example, `SyntacticObject::LexicalItemToken(...)` can be reduced to a much shorter macro invocation `so!(...)`. Similarly, LIs and LITs can be created with the macros `li!` and `lit!`

[3] To be precise, the Rust standard library does not define a set type called `Set`; rather, it defines two concrete implementations of a set type called `HashSet` and `BTreeSet`. `Set` is a type alias defined in `cands` that refers to `BTreeSet`.

respectively. Sets and vectors of features can be created with `fset!` and `fvec!`. The same code can be re-written more concisely as in Figure 2.

```
1 so!(lit!(li!(fset!( "M" );
2               fset!( "D" );
3               fvec![ "Mary" ]), 37))
```

Figure 2: Shorter code to create a SO.

An important feature of `cands` is the function `is_derivation`. This function implements the definition of derivations from C&S. It takes two arguments: `il`, of type `ILanguage`, which represents an I-language, and `stages`, of type `Vec<Stage>`, which represents a sequence of stages. `is_derivation(il, stages)` returns true iff `stages` is a derivation from `il` according to the definition in C&S.

We see two major usages of `cands`. First, it can be used to explore predictions from C&S. For example, one can check if a given sequence of stages is a valid derivation. Second, it can be extended to implement other notions and theories. C&S lacks formalization for many concepts that are popular in minimalist syntax, e.g., Agree, head movement and covert movement (Collins and Stabler, 2016). The predictions and empirical coverage of extensions to `cands` can be evaluated in a similar manner to the original `cands`.

3.2 Two wrapper programs for `cands`

Using `cands` requires programming in Rust, a relatively unfamiliar programming language among syntacticians. In order to make `cands` more accessible to the general audience, we provide two wrapper programs for `cands`. They are (a) `derivck`, a derivation checker that runs in the terminal, and (b) `derivexp`, an interactive derivation explorer that displays a GUI.

Figure 3 shows how the wrappers can be executed in a shell. Both programs require the user to provide an I-language IL and a sequence of stages S, both specified in JSON. These files are passed to the programs via command line arguments.

```
1 > derivck -i IL.json -d S.json
2 > derivexp -i IL.json -d S.json
```

Figure 3: Typical shell commands used to run `derivck` (line 1) and `derivexp` (line 2). The files specifying the I-language and the sequence of stages are passed via command line arguments.

`derivck` will output whether S is a derivation from IL. If not, `derivck` will display the offending stage(s) of S and a log that describes how it determined the stage(s) to be invalid. The log verbosity can be set with an environmental variable.

`derivexp` will first verify that S is a valid derivation. Then, it provides an interface that visualizes S and allows the user to apply various syntactic operations to the objects that comprise S to further advance the derivation. Figures 4a and 4b show screenshots from a `derivexp` session before and after the user has applied Merge to a pair of SOs.

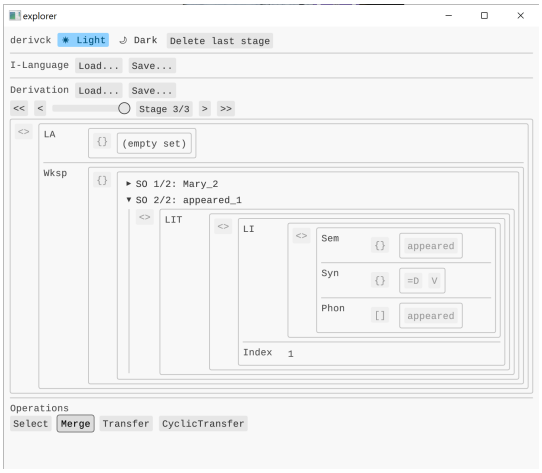
Both `derivck` and `derivexp` expect the JSON files for the I-language and the sequence of stages to be in a specific format that transparently reflects the Rust types for these two concepts, which are `ILanguage` and `Vec<Stage>`. This format is imposed by `serde`, a popular Rust data (de)serialization framework, which is used in `cands` to support human-readable JSON (de)serialization for its data structures. Even though we believe this format should be straightforward for users to follow, larger I-languages and sequences of stages in real-life use cases can be unwieldy to specify manually in JSON. In the near future, we plan to develop tools that would simplify the creation of these JSON files, such as a visual interface for constructing I-languages and sequences of stages and exporting them to JSON. For now, we provide sample JSON files in the Git repository for CANDS that can be used to construct a derivation for the simple sentence *Mary appeared*, as illustrated in Figure 4.[4]

We hope that `derivck` and `derivexp` will be useful for syntacticians working with the C&S system. If one already has a derivation in mind, they can check the derivation with `derivck`. Otherwise, one can use `derivexp` to explore the possible derivations generated by the C&S system. The two programs should facilitate working with grammatical and ungrammatical examples respectively.

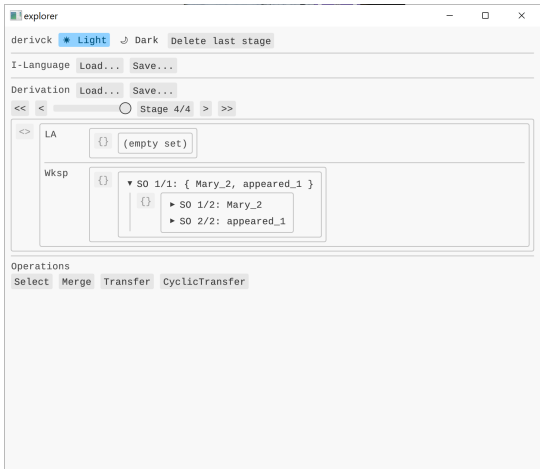
4 Evaluating theories with `cands`

An important and challenging task for syntacticians is to keep track of the empirical coverage of the syntactic theory at hand as one proposes changes to the theory. Often, one proposes a revision to the theory in order to make a correct prediction for one

[4]We thank one reviewer for pointing out the necessity to address how easily these JSON files can be created.



(a) `derivexp` is showing a stage S_3, whose workspace W_3 contains two roots: *Mary*$_2$ and *appeared*$_1$. We then apply Merge(*appeared*$_1$, *Mary*$_2$) to derive the next stage.



(b) We advance to the next stage S_4, whose workspace W_4 contains just one root, which is the result of Merge(*appeared*$_1$, *Mary*$_2$). `derivexp` is showing S_4.

Figure 4: Screenshots from a `derivexp` session.

sentence, only to realize later that another sentence correctly predicted by the old theory receives an incorrect prediction under the new theory.

Computational implementation of syntactic theories facilitates the process of examining their predictions and evaluating their empirical coverage. Using the function `is_derivation` defined in `cands`, it is easy to check if some derivation of interest can be generated by the C&S system. Even if one modifies `cands` in order to implement their revisions of C&S, predictions can be studied in the same way as long as the `is_derivation` function is preserved. Multiple revisions to C&S can be evaluated in terms of their empirical coverage by testing the corresponding modified versions of `cands` on a common set of derivations.

In this section, we illustrate this evaluation process with a simple example as a proof of concept. We consider the sentences in (1) and provide a derivation for each sentence. The original C&S system generates all three derivations, which is not ideal – we expect a good theory to only generate the derivations for the grammatical sentences. We will provide two attempts at positing a new constraint and incorporating it into C&S to correct the predictions. We will implement the new constraints as extensions of `cands`, and test these extensions on our derivations of interest. We will see that both attempts are inadequate in that each constraint fixes the prediction for one sentence while breaking the prediction for another. Our examples and analyses are inspired by classic literature on PP extraposition (Akmajian, 1975; Guéron, 1980; Wexler and Culicover, 1980).[5] For space reasons, we will only define the constraints and discuss their predictions conceptually in the main paper. We provide pseudocode for the implementations of these constraints in Appendix B, and the implementations themselves in the Git repository on the branches `theory1` and `theory2`.

(1) a. * A story bothered me about Mary.
b. A story appeared about Mary.
c. * I know who a story appeared about.

In (1a), PP extraposition occurs from the subject of a transitive verb. In (1b), the extraposition occurs from the subject of an unaccusative verb. In (1c), the same extraposition found in (1b) occurs, as well as *wh*-movement to embedded [Spec; CP].

The original C&S system allows for all three sentences to be derived, with the derivations sketched in (2), (3) and (4),[6] and fully detailed in Appendix C. To accommodate rightward extraposition in the LCA-like linearization algorithm employed in C&S, we use two covert heads X and Y as well as remnant movement. For example, in (2), X first merges with TP. The extraposed PP then moves to [Spec; XP]. Y then merges with XP, and the remnant TP moves to [Spec; YP]. Y contains the

[5] We thank Kyle Johnson for introducing us to the debate on PP extraposition when we were in search of syntactic phenomena to illustrate the usage of `cands` with.

[6] Non-final occurrences of SOs are struck out. Although the SOs in these derivations are actually sets, which should be denoted with comma-separated lists of elements enclosed in braces, we use the labelled bracket notation here to save space.

syntactic feature [T], which allows C to merge with YP as it would merge with a TP. In (2), the extraposition occurs from TP, while in (3) and (4) the extraposition occurs from VP.

Note on notations: we write $A \in^+ B$ for "A is contained in B", and $A \in^* B$ for "A is equal to or contained in B".

4.1 Theory 1: derivational constraint

Consider the pair (2) and (3). They differ in their grammaticality as well as the source of extraposition: the subject in the former derivation and the VP in the latter. We can predict these derivations correctly if we use a derivational flavor of the Subject Condition, i.e., a constraint that bans movement out of [Spec; TP]. Let us write $\mathrm{occ}_R(X)$ for the set of all occurrences of X in R. Then, we can add the condition (5) to the derive-by-Merge condition.

(5) **Derivational Subject Condition (DSC)**
For (internal) $\mathrm{Merge}(A, B)$ where A is the head and $B \in^+ A$, then $|\mathrm{occ}_A(B)| > \sum_{X \in \mathrm{Sbjs}_A(B)} |\mathrm{occ}_A(X)|$, where $\mathrm{Sbjs}_A(B)$ is the set of all [Spec; TP]s in A that contain B.

Consider internal $\mathrm{Merge}(A, B)$ where $B \in^+ A$. DSC holds iff there exists some occurrence B_P of B in A that is not equal to or contained in any occurrence of some [Spec; TP] in A. Thus, DSC holds iff this instance of Merge could be interpreted as movement from a non-subject position.

We call the C&S system extended by DSC "Theory 1." We implement and test Theory 1 against our derivations. The results show that only (2) is ungrammatical, so Theory 1 makes an incorrect prediction for (4). The PP extraposition in (2) violates DSC because all occurrences of the PP prior to this extraposition are contained under some occurrence of DP, which is at [Spec; TP]. The extrapositions in (3) and (4) do not violate DSC because the extraposition occurs before TP is even built. The subsequent *wh*-movement in (4) does not violate DSC either, due to the occurrence of *who* contained in the extraposed PP at [Spec; XP].

4.2 Theory 2: representational constraint

Consider the pair (2) and (4). They are both ungrammatical, and in both derivations there is a SO that has one occurrence inside and another occurrence outside of the subject, namely the PP *about Mary/who*. This suggests that perhaps the Subject Condition should be representational after all; any SO that has an occurrence inside some [Spec; TP] cannot have an occurrence outside that [Spec; TP]. This condition, formally stated as (6), is enforced at every stage of the derivation, applying to every workspace W_i.

Derivations	Truth	Theory 1	Theory 2
(2), for (1a)	*	*	*
(3), for (1b)	✓	✓	*
(4), for (1c)	*	✓	*

Table 1: Derivations, grammaticalities and predictions.

(6) **Representational Subject Condition (RSC)**
For any root $R \in W_i$ and any SOs $X, S \in^* R$ such that $X \in^* S$ and S is [Spec; TP], $|\mathrm{occ}_R(X)| = |\mathrm{occ}_R(S)|$.

If $X \in^* S \in^* R$, then every occurrence of S in R is either equal to or contains some occurrence of X in R (Theorem 1 from C&S). Thus $X \in^* S \in^* R$ implies $|\mathrm{occ}_R(X)| \geq |\mathrm{occ}_R(S)|$. If $|\mathrm{occ}_R(X)| > |\mathrm{occ}_R(S)|$, it must be the case that some occurrence of X is not equal to or contained in any occurrence of S. This is exactly the situation that RSC bans.

Let us call the C&S system extended by RSC "Theory 2". We implement and test Theory 2 against our derivations. Although Theory 2 correctly rules out (2) and (4), it incorrectly rules out (3) as well. This is because at the final stage in all three derivations, the PP *about Mary/who* has four occurrences, while the [Spec; TP] *a story about Mary/who*, which contains the PP, has three occurrences.

Table 1 summarizes the derivations, their desired grammaticalities and the grammaticalities predicted by our theories.

5 Extending theories with `cands`

In the literature, minimalist syntactic theories are usually described in text, with various degrees of formality. As such, it can be difficult to communicate the precise details of the theories to the reader. The benefit of implementing theories in code is that one is forced to consider and specify such details, because otherwise one would end up with an incomplete implementation.

Since C&S is a formalization of a bare-bones Minimallist syntactic theory, we expect that `cands` will provide a good starting point for min-

(2) a. Build TP.
[TP [DP a story [PP about Mary]] T bothered me]
b. Extrapose PP.
[YP [TP [DP a story ~~[PP about Mary]~~] T bothered me] Y [XP [PP about Mary] X ~~TP~~]]

(3) a. Build VP.
[VP appeared [DP a story [PP about Mary]]]
b. Extrapose PP.
[YP [VP appeared [DP a story ~~[PP about Mary]~~]] Y [XP [PP about Mary] X ~~VP~~]]
c. Build TP; move DP.
[TP [DP a story ~~PP~~] T [YP [VP appeared ~~DP~~] Y [XP [PP about Mary] X ~~VP~~]]]

(4) a. Same with (3) up to (3c), except we have *who* instead of *Mary*.
[TP [DP a story ~~PP~~] T [YP [VP appeared ~~DP~~] Y [XP [PP about who] X ~~VP~~]]]
b. Build CP; move *who*.
[CP who Q [TP [DP a story ~~PP~~] T [YP [VP appeared ~~DP~~] Y [XP [PP about ~~who~~] X ~~VP~~]]]]

imalist syntacticians to implement their own proposals and theories on top of it. To illustrate this, we implement two proposals for Agree, a syntactic operation commonly assumed by minimalist syntacticians but is undefined in C&S. Specifically, we implement two proposals, described respectively in Chomsky 2001 and Collins 2017. Our implementations can be found on the Git repository on branches `agree-chomsky-2001` and `agree-collins-2017`. We recognize that there are many other proposals for Agree, such as Pesetsky and Torrego 2007, Béjar and Rezac 2009, Zeijlstra 2012, Preminger 2014 and Deal 2015.

5.1 Agree à la Chomsky (2001)

First, we formalize and implement Chomsky's (2001) proposal for Agree.

Our system distinguishes two kinds of syntactic features: *normal syntactic features*, which are just like semantic and phonological features; and *valuable syntactic features*, which are associated with interpretability and a potential value.

(7) A syntactic feature is either normal or valuable.

a. A *normal syntactic feature* is some $F \in \text{SYN-F}$.

b. A *valuable syntactic feature* is some $F = \langle i, f, v \rangle$ where $i \in \{\text{i}, \text{u}\}$ is its *interpretability*, $f \in \text{SYN-F}$, and either $v = \_$ (*unvalued*) or $v = v'$ for some value v' (*valued*). F is usually denoted $[if{:}v]$ (e.g. [uCase:_], [iPerson:3]).

Agree is a function that takes two LITs, which we call the *probe* and the *goal*. The probe is valued with the features from the goal, and if the probe is not defective, the goal is valued with the features from the probe. Agree returns the new probe and the new goal.

(8) For lexical item tokens

$$P = \langle \langle \text{SEM}_P, \text{SYN}_P, \text{PHON}_P \rangle, k_P \rangle,$$
$$G = \langle \langle \text{SEM}_G, \text{SYN}_G, \text{PHON}_G \rangle, k_G \rangle,$$

$\text{Agree}(P, G) = \langle P', G' \rangle$ where

$$P' = \langle \langle \text{SEM}_P, \text{SYN}_{P'}, \text{PHON}_P \rangle, k_P \rangle,$$
$$G' = \langle \langle \text{SEM}_G, \text{SYN}_{G'}, \text{PHON}_G \rangle, k_G \rangle,$$

where

$$\text{SYN}_{P'} = \{\text{Value}(F, \text{SYN}_G) \mid F \in \text{SYN}_P\},$$
$$\text{SYN}_{G'} = \text{SYN}_G \text{ if } P \text{ is defective, otherwise}$$
$$= \{\text{Value}(F, \text{SYN}_P) \mid F \in \text{SYN}_G\}.$$

(9) For a syntactic feature F and a set of syntactic features SYN, $\text{Value}(F, \text{SYN}) =$

a. F, if F is normal or valued.

b. $\langle i, f, v' \rangle$, if $F = \langle i, f, v \rangle$ with $v = \_$, and there is $F' = \langle i', f', v' \rangle \in \text{SYN}$.

We modify Clause (iii) of the C&S definition of derivations by adding the derive-by-Agree condition, which checks if a workspace W_{i+1} can be derived from the previous workspace W_i by applying Agree to an appropriate probe-goal pair.

(10) (derive-by-Agree) Consider the ith workspace W_i. Fix some $R \in W_i$ and some active, matching pair of lexical item tokens P, G such that

a. P c-commands G, and

b. for any lexical item token $H \in^* R$ such that H matches P and P c-commands H, either $G = H$ or G c-commands H.

Let $\langle P', G' \rangle = \text{Agree}(P, G)$, and let $X = R$, except all occurrences of P and G are respectively replaced with P' and G'. Then the next workspace W_{i+1} is *derived-by-Agree* from W_i if $W_{i+1} = (W \setminus \{R\}) \cup \{R'\}$, where either

a. $R' = X$ and P doesn't contain the EPP-feature, or

b. $R' = \text{Merge}(X, Y)$ and P contains the EPP-feature, with some Y that satisfies $G \in^* Y \in^* X$ determined by pied-piping.

Derivation by Agree necessarily changes SOs in place, thereby violating the No-Tampering Condition (NTC; Chomsky 2007). As a result, upon finding an appropriate probe-goal pair, our implementation of derive-by-Agree visits the entire structure of R in order to construct X from R by replacing the old probe and goals with new ones.

During the construction of X, it is necessary to replace all occurrences of the goal G with the new goal G', rather than just replacing the highest occurrence. This is common practice in a multidominance-based theory like C&S. Otherwise, the highest occurrence of the post-Agree goal will no longer be considered as the same SO as the remaining occurrences, which has consequences in linearization.

We illustrate our implementation with (11), a derivation for the sentence *The man falls*.[7] The full derivation is in Appendix D.

5.2 Agree à la Collins (2017)

Next, we formalize and implement Collins' (2017) proposal for Agree. This proposal differs from Chomsky 2001 in two important ways: (a) Agree is not its own syntactic operation, but rather a special case of Merge, and (b) derivation by "Agree" complies with the NTC and does not modify SOs in-place; rather, features are Merged to feature-checking positions.

[7] π is Person, # is Number and C is Case.

As with our implementation of Chomsky 2001, we split syntactic features into *normal syntactic features* and *valuable syntactic features*. In this implementation, however, the value of valuable syntactic feature is required. Unlike Chomsky's feature valuation system, Collins's feature checking system does not allow features to be unvalued.

(13) A syntactic feature is either normal or valuable.

a. A *normal syntactic feature* is some $F \in \text{SYN-F}$.

b. A *valuable syntactic feature* is some $F = \langle i, f, v \rangle$ where $i \in \{\text{i}, \text{u}\}$ is its interpretability, $f \in \text{SYN-F}$, and v is some value.

We redefine SOs so that they can be created by Merging a SO and a syntactic feature.[8]

(14) X is a syntactic object iff

a. X is a lexical item token, or

b. $X = \text{Cyclic-Transfer}(\text{SO})$ for some syntactic object SO, or

c. X is a set of syntactic objects, or

d. $X = \{\text{SO}, F\}$ for some syntactic object SO and syntactic feature F.

As we redefine SOs, we must also change many definitions that depend on SOs. A crucial example is Triggers; just as some Triggers function T is able to check a feature off a SO if it is Merged with another approriate SO, T should able to check an uninterpretable feature off a SO if it is Merged with an appropriate syntactic feature. We change Clause (ii) in the definition of Triggers that handles SOs of the type $\{\text{SO}, F\}$:

(15) (ii) If $A = \{B, F\}$ where B is a SO, F is a syntactic feature and $\text{Triggers}(B) \neq \varnothing$, then $\text{Triggers}(A) = \text{Triggers}(B) \setminus \{\text{uF}\}$ for some uninterpretable syntactic feature $\text{uF} \in \text{Triggers}(B)$.

There are two cases of Merge we must consider: $\text{Merge}(A, B)$ where A, B are both SOs, and $\text{Merge}(A, F)$ where A is an SO and F is a syntactic feature. The first case is the old Merge, which we call Merge_{SO} from now on. The second case is Merge_{F}, which we define as follows:

[8] An alternative we do not explore in this paper is to allow syntactic features themselves be SOs.

(11) a. Build TP.
PRES has SYN = { [T], [=*v*], [EPP], [uπ:_], [u#:_], [iC:nom] }.
man has SYN = { [N], [iπ:3], [i#:sg], [uC:_] }.
$W_i = \{\{$ PRES, { *v*, { falls, { the, man }}}}}

b. Agree applies, with PRES as the probe and *man* as the goal. They are replaced with PRES′ and *man*′. Since PRES has EPP, the DP *the man*′ is pied-piped to [Spec; TP].
PRES′ has SYN = { [T], [=*v*], [EPP], [uπ:3], [u#:sg], [iC:nom] }.
man′ has SYN = { [N], [iπ:3], [i#:sg], [uC:nom] }.
$W_j = \{$ { the, man′ }, { PRES′, { *v*, { falls, { the, man′ }}}}}

(12) a. Select PRES and *man*.
PRES has SYN = { [T], [=*v*], [EPP], [uπ:3], [u#:sg], [iC:nom] }.
man has SYN = { [N], [iπ:3], [i#:sg], [uC:nom] }.
$W_i = \{$PRES, man$\}$

b. Merge *man* with [iC:nom] from PRES.
$W_j = \{$PRES, { man, [iC:nom] }$\}$

c. Build TP, up to and including movement of *the man* to [Spec; TP]. Call the result TP_k.
$W_k = \{\underbrace{\{ \{ \text{the}, \{ \text{man}, [\text{iC:nom}] \} \}, \{ \text{PRES}, \{ v, \{ \text{falls}, \{ \text{the}, \{ \text{man}, [\text{iC:nom}] \} \}\}\}\}\}}_{\text{TP}_k}\}$

d. Merge TP_k with [iπ:3], then with [i#:sg], both from *man*.
$W_\ell = \{\{$ [i#:sg], { [iπ:3], TP_k }}$\}$

(16) Given any syntactic object X and syntactic feature F, where $\text{Triggers}(X) \neq \varnothing$, $\text{Merge}_\text{F}(X, F) = \{X, F\}$.

Finally, we modify Clause (iii) from the definition of derivations. The derive-by-Merge condition must be split in two cases: derive-by-Merge_SO, which is the old derive-by-Merge, and derive-by-Merge_F, which handles derivation by $\text{Merge}_\text{F}(A, F)$ for some SO A and syntactic feature F. Derive-by-Merge_F requires F to be part of some LIT contained in the workspace, but not necessarily contained in A. In other words, sideward Merge is allowed only for Merge_F.

(17) (derive-by-Merge_F) $\text{LA}_i = \text{LA}_{i+1}$ and the following conditions hold for some A, F:
a. $A \in W_i$,
b. There exists some lexical item token $X \in^+ W_i$ such that $X = \langle\langle \text{SEM}, \text{SYN}, \text{PHON}\rangle, k\rangle$ where $F \in \text{SYN}$, and
c. $W_{i+1} = (W_i \backslash \{A\}) \cup \{\text{Merge}_\text{F}(A, F)\}$.

We illustrate our implementation with (12), a derivation for the sentence *The man falls*. This derivation is based on Derivation (27) in Collins 2017, where the T head PRES Merges with the ϕ-features from *man* to form the complex T { PRES, [iϕ] } before Merging with *v*P. This is problematic, as $\text{Transfer}_\text{PF}$ cannot linearize the TP { { PRES, [iϕ] }, *v*P} because *v*P is neither a complement, as the complex T is not a LIT; nor is *v*P a specifier, as the complex T is not a set of SOs either. In our derivation (12), we let PRES Merge with its *v*P complement before Merging with the ϕ-features, avoiding the $\text{Transfer}_\text{PF}$ problem. The full derivation is in Appendix D.

6 Future work

In Section 4, we showed how extensions of `cands` can be evaluated against a common set of derivations, offering a quantitative comparison of their empirical coverages. Our evaluation setup can be scaled up quite easily, by curating a large-scale test set of derivations, which can then be used to evaluate `cands`-based implementations of many different theories. This kind of evaluation is familiar in the parsing literature, where parsers are evaluated on large datasets of syntactically annotated sentences known as *treebanks*, such as the Penn Treebank (Marcus et al., 1993), CCGbank (Hockenmaier and Steedman, 2002), the Redwoods treebank (Flickinger, 2011; Oepen et al., 2002, 2004), MGbank (Torr, 2017, 2018), among others.

While `cands` can check if C&S generates a given derivation, it cannot check if C&S generates some derivation that linearizes to a given PF. Obviously, syntacticians are equally if not more interested in problems of the latter type. For example, one might wish to check if a theory overgenerates, i.e., if it derives an ungrammatical sentence, or if it derives a grammatical sentence with an undesirable derivation. Solving this type of problems requires us to develop an algorithm that automatically explores the predictions from C&S, which is essentially a parser. There is a recent line of work on neural transition-based parsers, i.e. neural classifiers that take parser states as input and output parser transitions as output (Dyer et al., 2016; Yoshida and Oseki, 2022; Sartran et al., 2022). While these parsers are typically implemented with state-of-the-art neural architectures, they usually only support parsing for primitive grammars, such as PCFGs. As such, we hope to explore if neural transition-based parsers can be developed for more complex grammars, such as Minimalist Grammars (Stabler, 1997) and C&S. An even more challenging task is to develop methods to (semi)automatically derive a parser for an arbitrary extension of C&S.

Finally, `cands` brings us closer to the quantitative evaluation of the parsimony of C&S and relevant theories. For example, any `cands`-based implementation of some theory provides an upper bound for the *minimum description length (MDL)* of that theory. MDL can in turn be used to define a prior distribution over theories in a probabilistic setup (Berwick, 2015).

7 Conclusion

We present CANDS, a Rust implementation of Collins and Stabler's (2016; C&S) formalization of a minimalist syntactic theory. The core of CANDS is `cands`, a library. `cands` by itself can be used to explore predictions from the C&S system, and it can also be extended to implement other theoretical notions. We also present `derivck` and `derivexp`, two wrapper programs that allows the user to check and explore derivations with `cands` without having to program in Rust.

Computational implementation of syntactic theories greatly facilitates the evaluation of their empirical coverages, and forces the programmer to attend to the details and edge cases of the theories, which can be easily miscommunicated in textual descriptions of minimalist syntactic theory. In this paper, we show how CANDS can be integrated into a minimalist syntactician's typical workflow. We hope our work will benefit the minimalist syntax community, and we welcome suggestions and contributions, as our work is still under much development.

Acknowledgments

We would like to thank Yushi Sugimoto and other members of OsekiLab at the University of Tokyo for their helpful comments and suggestions at various stages of this project. We also want to thank Kyle Johnson, Faruk Akkuş, and three anonymous reviewers for their thoughtful feedback. All remaining errors are ours. This work was supported by JST PRESTO Grant Number JPMJPR21C2.

References

Adrian Akmajian. 1975. More evidence for an np cycle. *Linguistic Inquiry*, 6(1):115–129.

Susana Béjar and Milan Rezac. 2009. Cyclic agree. *Linguistic Inquiry*, 40(1):35–73.

Emily M Bender, Scott Drellishak, Antske Fokkens, Michael Wayne Goodman, Daniel P Mills, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar prototyping and testing with the lingo grammar matrix customization system. In *Proceedings of the ACL 2010 system demonstrations*, pages 1–6.

Emily M Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *COLING-02: Grammar Engineering and Evaluation*.

Emily M Bender, Dan Flickinger, and Stephan Oepen. 2008. Grammar engineering for linguistic hypothesis testing. In *Proceedings of the Texas Linguistics Society X Conference: Computational linguistics for less-studied languages*, pages 16–36.

Robert C. Berwick. 2015. Mind the gap. In Ángel J Gallego and Dennis Ott, editors, *50 years later: Reflections on Chomsky's Aspects*. MIT Working Papers in Linguistics.

Manfred Bierwisch. 1963. *Grammatik des deutschen Verbs*. Akademie Verlag, Berlin.

Miriam Butt. 1999. A grammar writer's cookbook. *CSLI Lecture Notes*.

Noam Chomsky. 1995. *The Minimalist Program*. MIT press.

Noam Chomsky. 2001. Derivation by Phase. In *Ken Hale: A Life in Language*. The MIT Press.

Noam Chomsky. 2007. Approaching ug from below. *Interfaces+ recursion= language*, 89:1–30.

Chris Collins. 2017. Merge (x, y)={X, Y}. In Leah Bauke and Andreas Blümel, editors, *Labels and roots*, pages 47–68. De Gruyter Mouton.

Chris Collins and Edward Stabler. 2016. A formalization of minimalist syntax. *Syntax*, 19(1):43–78.

Amy Rose Deal. 2015. Interaction and satisfaction in φ-agreement. In *Proceedings of NELS*, volume 45, pages 179–192.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Dan Flickinger. 2011. Accuracy vs. robustness in grammar engineering. *Language from a cognitive perspective: Grammar, usage, and processing*, 201:31–50.

Antske Sibelle Fokkens. 2014. Enhancing empirical research for linguistically motivated precision grammars. *Saarbrücken: Saarland University*.

Sandiway Fong and Jason Ginsburg. 2019. 16Towards a Minimalist Machine. In *Minimalist Parsing*. Oxford University Press.

John Frampton. 2004. Copies, traces, occurrences, and all that. *Ms., Northeastern University*.

Jacqueline Guéron. 1980. On the syntax and semantics of pp extraposition. *Linguistic inquiry*, 11(4):637–678.

Julia Hockenmaier and Mark Steedman. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *LREC*, volume 42, page 58.

Marcus Kracht. 1999. Adjunction structures and syntactic domains. In Hans-Peter Kolb and Uwe Mönnich, editors, *The Mathematics of Syntactic Structure: Trees and their Logics*. De Gruyter Mouton, Berlin, Boston.

Marcus Kracht. 2001. Syntax in chains. *Linguistics and Philosophy*, 24:467–530.

Marcus Kracht. 2008. On the logic of lgb type structures. part i: Multidominance structures. *Logics for linguistic structures*, pages 105–142.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Stefan Müller. 2015. The coregram project: theoretical linguistics, theory development, and verification. *Journal of Language Modelling*, 3(1):21–86.

Stefan Müller. 1999. *Deutsche Syntax deklarativ*. Max Niemeyer Verlag, Berlin, Boston.

Stephan Oepen, Emily M Bender, Uli Callmeier, Dan Flickinger, and Melanie Siegel. 2002. Parallel distributed grammar engineering for practical applications. In *COLING-02: Grammar Engineering and Evaluation*.

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D Manning. 2004. Lingo redwoods: A rich and dynamic treebank for hpsg. *Research on Language and Computation*, 2:575–596.

David Pesetsky and Esther Torrego. 2007. The syntax of valuation and the interpretability of features. *Phrasal and clausal architecture*, pages 262–294.

Omer Preminger. 2014. *Agreement and Its Failures*. MIT Press.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439.

Edward Stabler. 1997. Derivational minimalism. In *Logical Aspects of Computational Linguistics: First International Conference, LACL'96 Nancy, France, September 23–25, 1996 Selected Papers 1*, pages 68–95. Springer.

John Torr. 2017. Autobank: a semi-automatic annotation tool for developing deep minimalist grammar treebanks. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 81–86.

John Torr. 2018. Constraining mgbank: Agreement, l-selection and supertagging in minimalist grammars. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 590–600.

Mettina Jolanda Arnoldina Veenstra. 1998. *Formalizing the minimalist program*. Ph.D. thesis, Rijksuniversiteit Groningen.

Kenneth Wexler and Peter W Culicover. 1980. *Formal principles of language acquisition*, volume 76. MIT press Cambridge, MA.

Ryo Yoshida and Yohei Oseki. 2022. Composition, attention, or both? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5822–5834, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Olga Zamaraeva. 2021. *Assembling syntax: Modeling constituent questions in a grammar engineering framework*. University of Washington.

Olga Zamaraeva, Chris Curtis, Guy Emerson, Antske Fokkens, Michael Wayne Goodman, Kristen Howell, TJ Trimble, and Emily M Bender. 2022. 20 years of the grammar matrix: cross-linguistic hypothesis testing of increasingly complex interactions. *Journal of Language Modelling*, 10(1):49–137.

Hedde Zeijlstra. 2012. There is only one way to agree. *The linguistic review*, 29(3):491–539.

Arnold M Zwicky, Joyce Friedman, Barbara C Hall, and Donald E Walker. 1965. The mitre syntactic analysis procedure for transformational grammars. In *Proceedings of the November 30–December 1, 1965, fall joint computer conference, part I*, pages 317–326.

A List of C&S definitions that are implemented in `cands`

Table 2 contains a list of all definitions in C&S. For each definition, we indicate whether it is implemented in `cands`.

We left four groups of definitions from C&S unimplemented. The first group consists of tentative definitions; they are presented in earlier parts of the C&S paper, and eventually replaced by more complete definitions later in the paper. Specifically, this group consists of Definitions 8 (SO) and 14 (derivation), which are replaced by Definitions 37 and 38. We implement the latter definitions instead of the former ones.

The second group of unimplemented definitions simply cannot be implemented. This applies to Definitions 15, 15′ and 23. These define the concept of the derivability of a given SO or workspace. Derivability itself is a binary value, either true or false – it is a trivial definition that does not need an implementation. Presumably, it is more interesting to implement a function that would compute the derivability from a given SO or workspace. To implement such a function, we need to create a parser for the C&S system. This is beyond the scope of our paper.

The third group of unimplemented definitions are unnecessary to implement. This applies to Definition 25, which defines trigger features. Trigger features are just a special name to designate a certain group of features for a particular $\mathrm{Triggers}$ implementation. As the concept is purely expository, it has no place in our implementation of C&S.

The last group of unimplemented definitions concern occurrences (Definitions 16, 17, 18, 20, 22) and chain-based SOs (Definitions 16′, 7′, 13′, 14′, 15′), which are only partially explored in C&S as a digression from their full formalization of a theory of token-based SOs. We leave their implementations to future work.

No.	Definition	In `cands`?
Section 2: Preliminary definitions		
1	Universal Grammar	Yes
2	Lexical item	Yes
3	Lexicon	Yes
4	I-language	Yes
5	Lexical item token	Yes
6	Lexical array	Yes
7	Syntactic object (old)	No
8	Immediate containment (SO)	Yes
9	Containment	Yes
Section 3: Workspaces, Select, and Merge		
10	Stage	Yes
	Workspace	Yes
11	Roothood	Yes
12	Select	Yes
13	Merge	Yes
14	Derivation (old)	No
15	Derivability from lexicon	No
Section 4: Occurrences		
16	Position	No
17	Occurrence	No
18	Immediate containment (occurrence)	No
19	Sisterhood (SO)	Yes
20	Sisterhood (occurrence)	No
21	C-command (SO)	Yes
	Asymmetric c-command (SO)	Yes
22	C-command (occurrence)	No
Section 5: Digression		
$16'$	Path (chain-based)	No
$7'$	SO (chain-based)	No
$13'$	Merge (chain-based)	No
$14'$	Derivation (chain-based)	No
$15'$	Derivability from lexicon (chain-based)	No

No.	Definition	In `cands`?
Section 6: General Properties of Derivations		
23	Derivability	No
24	Binary branching	Yes
Section 7: Labels		
25	Trigger feature	No
26	Triggers	Yes
27	Triggered Merge	Yes
28	Label	Yes
29	Maximal projection	Yes
30	Minimal projection	Yes
31	Intermediate projection	Yes
32	Complement	Yes
33	Specifier	Yes
Section 8: Transfer		
34	Transfer	Yes
35	Strong phasehood	Yes
36	Cyclic-Transfer	Yes
37	Syntactic object (new)	Yes
38	Derivation (new)	Yes
Section 9: $\text{Transfer}_{\text{LF}}$		
39	$\text{Transfer}_{\text{LF}}$	Yes
Section 10: $\text{Transfer}_{\text{PF}}$		
40	Finality	Yes
41	$\text{Transfer}_{\text{PF}}$	Yes
Section 13: Convergence		
42	Convergence and crash at the CI interface	Yes
43	Convergence and crash at the SM interface	Yes
44	Convergence and crash	Yes

Table 2: List of definitions in C&S. For each definition, we indicate whether it is implemented in `cands`.

B Implementing the extensions of `cands` for PP extraposition

In Section 4, we described two extensions of C&S, where each extension is created by adding one constraint into the C&S system. In this section, we describe our implementation of these extensions in more detail.

B.1 Theory 1

Theory 1 is the extension of C&S by the Derivational Subject Condition (DSC), defined in (5). DSC further constricts the derive-by-Merge condition, specifically the internal Merge case.

The derive-by-Merge condition is checked by the `derive_by_merge` function, which is used by the `is_derivation` to check if each non-initial stage is derived from its previous stage by an appropriate application of a syntactic operation, including Merge. We implement DSC inside the `derive_by_merge` function. The pseudocode for `derive_by_merge` as well as the DSC is provided in Algorithm 1. The for-loop starting on line 6 checks for internal Merge, and the for-loop starting on line 13 checks for external Merge. Once an appropriate pair of SOs A, B is found in either for-loop, the function returns true from within that loop. The DSC is thus implemented in the for-loop for internal Merge. At line 9, we check the negation of DSC; if the DSC is violated, the if-statement is executed, and the current iteration of the for-loop will be skipped (also known as a *continue*-statement). As such, the return-statement on line 12 is unreachable in the current iteration. This implements the DSC.

B.2 Theory 2

Theory 2 is the extension of C&S by the Representational Subject Condition (RSC), defined in (6). RSC is checked for every stage in the derivation.

We implement RSC is the `is_derivation` function, whose pseudocode is provided in Algorithm 2. The for-loop starting on line 7 checks whether each pair of consecutive stages is derived-by-Select, Merge or Transfer. The if-statement on line 8 checks if neither of these three syntactic operations derive the second stage from the first, in which case the function returns false. RSC further constraints this check. If RSC is violated, the if-statement on line 15 will execute, and the function returns false.

Input: Two stages $S_1 = \langle \mathrm{LA}_1, W_1 \rangle$ and $S_2 = \langle \mathrm{LA}_2, W_2 \rangle$
Output: true iff S_2 is derived-by-Merge from S_1

```
1  if LA_1 ≠ LA_2 then
2      return false;
3  if W_1 is empty then
4      return false;
5  foreach A ∈* W_1 do
6      foreach B such that B ∈* A do
           /* ===== DSC begins =====                                   */
7          Calculate |occ_A(B)| ;
8          Calculate Σ_X |occ_A(X)|, the sum of |occ_A(X)| for all [Spec; TP] X ∈* A such that
             B ∈* X ;
9          if |occ_A(B)| ≤ Σ_X |occ_A(X)| then
10             Skip to the next pair of A, B;
           /* ====== DSC ends ======                                   */
11         if W_2 = W_1 \ {A, B} ∪ {Merge(A, B)} then
12             return true;
13     foreach B such that B ∈ W_1 do
14         if W_2 = W_1 \ {A, B} ∪ {Merge(A, B)} then
15             return true;
16 return false;
```

Algorithm 1: Pseudocode for the `derive_by_merge` function. The implementation of DSC is between lines 7–10, inclusive on both ends.

Input: an I-language $\mathrm{IL} = \langle \mathrm{Lex}, \mathrm{UG} \rangle$, and a sequence of stages $S = \langle S_1, \cdots, S_n \rangle$, with $S_i = \langle \mathrm{LA}_i, W_i \rangle$ for each $i \in [n]$

Output: true iff S is a derivation from IL

```
1  if S is empty then
2      return false;
3  if there is some LIT X ∈ LA_1 that is not contained in Lex then
4      return false;
5  if W_1 is not empty then
6      return false;
7  foreach i < n do
8      if S_{i+1} is not derived-by-Select from S_i and S_{i+1} is not derived-by-Merge from S_i and S_{i+1} is
         not derived-by-Transfer from S_i then
9          return false;
       /* ===== RSC begins =====                                          */
10     foreach R ∈ W_{i+1} do
11         let S = the set of all S ∈* R such that S is [Spec; TP];
12         let X = the set of all X ∈* S for some S ∈ S;
13         Calculate |occ_R(S)| for each S ∈ S ;
14         Calculate |occ_R(X)| for each X ∈ X ;
15         if |occ_R(X)| ≠ |occ_R(S)| for any S ∈ S and any X ∈ X then
16             return false;
       /* ====== RSC ends ======                                          */
17 return true;
```

Algorithm 2: Pseudocode for the `is_derivation` function. The implementation of RSC is between lines 10–16, inclusive on both ends.

C Full derivations for the extraposition sentences

Here, we provide the full derivations for the sentences (1a), (1b) and (1c) in Section 4. These derivations were sketched in the main paper as (2), (3) and (4).

We assume the lexicon in Table 3. The semantic, syntactic and phonological features of our UG are the unions of the semantic, syntactic and phonological features over the LIs in our lexicon. The syntactic features include (a) category features of the form [α], where α is a syntactic category; (b) selectional features of the form [=α], where α is a syntactic category; (c) EPP-feature [EPP], and (d) *wh*-features [u*wh*] and [i*wh*]. Selectional features, EPP-feature and [u*wh*] are trigger features. A selectional feature [=α] can be checked by Merging with some SO whose label bears the category feature [α]. An EPP-feature can be checked by Merging with some SO. [u*wh*] can be checked by Merging with some SO whose labels bears [i*wh*]. We use two pairs of heads X and Y to handle extraposition; we use $\mathrm{X_{T,P}}$ and $\mathrm{Y_T}$ to handle PP extraposition from TP and use $\mathrm{X_{V,P}}$ and $\mathrm{Y_V}$ to handle PP extraposition from VP.

The derivations (18), (19) and (20) are for the sentences (1a), (1b) and (1c) respectively. For each stage S_i, we describe the syntactic operation by which S_i is derived, and we show its workspace W_i. We omit Select for brevity. Transferred SOs are struck out.

(18)

a. $\mathrm{Merge}(\textit{bothered}, \textit{me})$.
$W_1 = \{\underbrace{\{\text{ bothered, me }\}}_{\mathrm{VP}}\}$.

b. $\mathrm{Merge}(v^*, \mathrm{VP})$.
$W_2 = \{\underbrace{\{\ v^*, \mathrm{VP}\ \}}_{v^*\mathrm{P}_1}\}$.

c. $\mathrm{Transfer}(v^*\mathrm{P}_1, \mathrm{VP})$.
$W_3 = \{\underbrace{\{\ v^*, \cancel{\mathrm{VP}}\ \}}_{v^*\mathrm{P}_2}\}$.

d. $\mathrm{Merge}(\textit{about}, \textit{Mary})$.
$W_4 = \{\underbrace{\{\text{ about, Mary }\}}_{\mathrm{PP}}, v^*\mathrm{P}_2\}$.

e. $\mathrm{Merge}(\textit{story}, \mathrm{PP})$.
$W_5 = \{\underbrace{\{\text{ story, PP }\}}_{\mathrm{NP}}, v^*\mathrm{P}_2\}$.

f. $\mathrm{Merge}(a, \mathrm{NP})$.
$W_6 = \{\underbrace{\{\text{ a, NP }\}}_{\mathrm{DP}}, v^*\mathrm{P}_2\}$.

g. $\mathrm{Merge}(v^*\mathrm{P}_2, \mathrm{DP})$.
$W_7 = \{\underbrace{\{\text{ DP, }v^*\mathrm{P}_2\ \}}_{v^*\mathrm{P}_3}\}$.

h. $\mathrm{Merge}(\mathrm{PAST}_{v^*}, v^*\mathrm{P}_3)$.
$W_8 = \{\underbrace{\{\ \mathrm{PAST}_{v^*}, v^*\mathrm{P}_3\ \}}_{\mathrm{TP}_1}\}$.

i. $\mathrm{Merge}(\mathrm{TP}_1, \mathrm{DP})$.
$W_9 = \{\underbrace{\{\text{ DP, TP}_1\ \}}_{\mathrm{TP}_2}\}$.

j. $\mathrm{Merge}(\mathrm{X_{T,P}}, \mathrm{TP}_2)$.
$W_{10} = \{\underbrace{\{\text{ X}_{\mathrm{T,P}}\text{, TP}_2\ \}}_{\mathrm{XP}_1}\}$.

k. $\mathrm{Merge}(\mathrm{XP}_1, \mathrm{PP})$.
$W_{11} = \{\underbrace{\{\text{ PP, XP}_1\ \}}_{\mathrm{XP}_2}\}$.

l. $\mathrm{Merge}(\mathrm{Y_T}, \mathrm{XP}_2)$.
$W_{12} = \{\underbrace{\{\text{ Y}_{\mathrm{T}}\text{, XP}_2\ \}}_{\mathrm{YP}_1}\}$.

m. $\mathrm{Merge}(\mathrm{YP}_1, \mathrm{TP}_2)$.
$W_{13} = \{\underbrace{\{\text{ TP}_2\text{, YP}_1\ \}}_{\mathrm{YP}_2}\}$.

n. $\mathrm{Merge}(\mathrm{C}, \mathrm{YP}_2)$.
$W_{14} = \{\underbrace{\{\text{ C, YP}_2\ \}}_{\mathrm{CP}}\}$.

o. $\mathrm{Transfer}(\mathrm{CP}, \mathrm{CP})$.
$W_{15} = \{\cancel{\mathrm{CP}}\}$.

(19) a. $\text{Merge}(\textit{about}, \textit{Mary})$.
$W_1 = \{\underbrace{\{\text{ about, Mary }\}}_{\text{PP}}\}$.

b. $\text{Merge}(\textit{story}, \text{PP})$.
$W_2 = \{\underbrace{\{\text{ story, PP }\}}_{\text{NP}}\}$.

c. $\text{Merge}(\textit{a}, \text{NP})$.
$W_3 = \{\underbrace{\{\text{ a, NP }\}}_{\text{DP}}\}$.

d. $\text{Merge}(\textit{appeared}, \text{DP})$.
$W_4 = \{\underbrace{\{\text{ appeared, DP }\}}_{\text{VP}}\}$.

e. $\text{Merge}(\text{X}_{\text{V,P}}, \text{VP})$.
$W_5 = \{\underbrace{\{\text{ X}_{\text{V,P}}\text{, VP }\}}_{\text{XP}_1}\}$.

f. $\text{Merge}(\text{XP}_1, \text{PP})$.
$W_6 = \{\underbrace{\{\text{ PP, XP}_1\text{ }\}}_{\text{XP}_2}\}$.

g. $\text{Merge}(\text{Y}_{\text{V}}, \text{XP}_2)$.
$W_7 = \{\underbrace{\{\text{ Y}_{\text{V}}\text{, XP}_2\text{ }\}}_{\text{YP}_1}\}$.

h. $\text{Merge}(\text{YP}_1, \text{VP})$.
$W_8 = \{\underbrace{\{\text{ VP, YP}_1\text{ }\}}_{\text{YP}_2}\}$.

i. $\text{Merge}(v, \text{YP}_2)$.
$W_9 = \{\underbrace{\{\text{ }v\text{, YP}_2\text{ }\}}_{v\text{P}}\}$.

j. $\text{Merge}(\text{PAST}_v, v\text{P})$.
$W_{10} = \{\underbrace{\{\text{ PAST}_v\text{, }v\text{P }\}}_{\text{TP}_1}\}$.

k. $\text{Merge}(\text{TP}_1, \text{DP})$.
$W_{11} = \{\underbrace{\{\text{ DP, TP}_1\text{ }\}}_{\text{TP}_2}\}$.

l. $\text{Merge}(\text{C}, \text{TP}_2)$.
$W_{12} = \{\underbrace{\{\text{ C, TP}_2\text{ }\}}_{\text{CP}}\}$.

m. $\text{Transfer}(\text{CP}, \text{CP})$.
$W_{13} = \{\cancel{\text{CP}}\}$.

(20) a. Same as (19) up to and including (19m), but replace *Mary* with *who*.
$W_{13} = \{\cancel{\text{CP}_1}\}$.

b. $\text{Merge}(\textit{know}, \text{CP}_1)$.
$W_{14} = \{\underbrace{\{\text{ know, CP}_1\text{ }\}}_{\text{VP}}\}$.

c. $\text{Merge}(v^*, \text{VP})$.
$W_{15} = \{\underbrace{\{\text{ }v^*\text{, VP }\}}_{v^*\text{P}_1}\}$.

d. $\text{Transfer}(v^*\text{P}_1, \text{VP})$.
$W_{16} = \{\underbrace{\{\text{ }v^*\text{, }\cancel{\text{VP}}\text{ }\}}_{v^*\text{P}_2}\}$.

e. $\text{Merge}(v^*\text{P}_2, \textit{we})$.
$W_{17} = \{\underbrace{\{\text{ we, }v^*\text{P}_2\text{ }\}}_{v^*\text{P}_3}\}$.

f. $\text{Merge}(\text{PRES}_{v^*}, v^*\text{P}_3)$.
$W_{18} = \{\underbrace{\{\text{ PRES}_{v^*}\text{, }v^*\text{P}_3\text{ }\}}_{\text{TP}_1}\}$.

g. $\text{Merge}(\text{TP}_1, \text{DP})$.
$W_{19} = \{\underbrace{\{\text{ DP, TP}_1\text{ }\}}_{\text{TP}_2}\}$.

h. $\text{Merge}(\text{C}, \text{TP}_2)$.
$W_{20} = \{\underbrace{\{\text{ C, TP}_2\text{ }\}}_{\text{CP}_2}\}$.

i. $\text{Transfer}(\text{CP}_2, \text{CP}_2)$.
$W_{21} = \{\cancel{\text{CP}_2}\}$.

LI	**SEM**	**SYN**	**PHON**
Mary	{[Mary]}	{[D]}	⟨[Mary]⟩
me	{[me]}	{[D]}	⟨[me]⟩
we	{[we]}	{[D]}	⟨[we]⟩
who	{[who]}	{[D], [i*wh*]}	⟨[who]⟩
about	{[about]}	{[P], [=D]}	⟨[about]⟩
story	{[story]}	{[N], [=P]}	⟨[story]⟩
a	{[a]}	{[D], [=N]}	⟨[a]⟩
bothered	{[bothered]}	{[V], [=D]}	⟨[bothered]⟩
appeared	{[appeared]}	{[V], [=D]}	⟨[appeared]⟩
know	{[know]}	{[V], [=C]}	⟨[know]⟩
*v**	{[*v**]}	{[*v**], [=V], [=D]}	⟨⟩
v	{[*v*]}	{[*v*], [=V]}	⟨⟩
$X_{T,P}$	{[X]}	{[X], [=T], [=P]}	⟨⟩
$X_{V,P}$	{[X]}	{[X], [=V], [=P]}	⟨⟩
Y_T	{[Y]}	{[T], [=X], [=T]}	⟨⟩
Y_V	{[Y]}	{[V], [=X], [=V]}	⟨⟩
PRES_{v*}	{[PRES]}	{[T], [=*v**], [EPP]}	⟨⟩
PAST_{v*}	{[PAST]}	{[T], [=*v**], [EPP]}	⟨⟩
PAST_{v}	{[PAST]}	{[T], [=*v*], [EPP]}	⟨⟩
C	{[C]}	{[C], [=T]}	⟨⟩
Q	{[Q]}	{[C], [=T], [u*wh*]}	⟨⟩

Table 3: Lexicon for the derivations (18), (19) and (20). For example, the LI *Mary* is a 3-tuple $\langle \text{SEM}, \text{SYN}, \text{PHON} \rangle$ where $\text{SEM} = \{[\text{Mary}]\}$, $\text{SYN} = \{[\text{D}]\}$ and $\text{PHON} = \langle [\text{Mary}] \rangle$.

D Full derivations for *The man falls*

Here, we provide the full derivations for the sentence *The man falls* in Section 5. These derivations were sketched in the main paper as (11) and (12).

We assume the lexicon in Table 4. Again, the semantic, syntactic and phonological features of our UG are the unions of the semantic, syntactic and phonological features over the LIs in our lexicon.

The derivations (21) and (22) are for the sentence *The man falls* in our implementations of Chomsky (2001) and Collins (2017) respectively. We omit most applications of Select for brevity, except at the beginning of (22). There, it is important that the tense head PRES be selected near the beginning of the derivation, before any Merge takes place.

LI	SEM	SYN	PHON
the	{[the]}	{[D], [=N]}	⟨[the]⟩
falls	{[falls]}	{[V], [=D]}	⟨[falls]⟩
v	{[*v*]}	{[*v*], [=V]}	⟨⟩
C	{[C]}	{[C], [=T]}	⟨⟩
Unique to our Chomsky 2001 implementation:			
man	{[man]}	{[N], [iπ:3], [i#:sg], [uC:_]}	⟨[man]⟩
man′	{[man]}	{[N], [iπ:3], [i#:sg], [uC:nom]}	⟨[man]⟩
PRES	{[PRES]}	{[T], [=*v*], [EPP], [uπ:_], [u#:_], [iC:nom]}	⟨⟩
PRES′	{[PRES]}	{[T], [=*v*], [EPP], [uπ:3], [u#:sg], [iC:nom]}	⟨⟩
Unique to our Collins 2017 implementation:			
man	{[man]}	{[N], [iπ:3], [i#:sg], [uC:nom]}	⟨[man]⟩
PRES	{[PRES]}	{[T], [=*v*], [EPP], [uπ:3], [u#:sg], [iC:nom]}	⟨⟩

Table 4: Lexicon for the derivations (21) and (22).

(21) a. Merge(*the*, *man*).
$W_1 = \{\underbrace{\{\text{ the, man }\}}_{\text{DP}}\}$.

b. Merge(*falls*, DP).
$W_2 = \{\underbrace{\{\text{ falls, }\{\text{ the, man }\}\ \}}_{\text{VP}}\}$.

c. Merge(v, VP).
$W_3 = \{\underbrace{\{\text{ }v\text{, }\{\text{ falls, }\{\text{ the, man }\}\ \}\ \}}_{v\text{P}}\}$.

d. Merge(PRES, vP).
$W_4 = \{\underbrace{\{\text{ PRES, }\{\text{ }v\text{, }\{\text{ falls, }\{\text{ the, man }\}\ \}\ \}\ \}}_{\text{TP}_1}\}$.

e. Agree(PRES, *man*).
$W_5 = \{\underbrace{\{\text{ }\{\text{ the, man}'\text{ }\}\text{, }\{\text{ PRES}'\text{, }\{\text{ }v\text{, }\{\text{ falls, }\{\text{ the, man}'\text{ }\}\ \}\ \}\ \}\ \}}_{\text{TP}_2}\}$.

f. Merge(C, TP_2).
$W_6 = \{\underbrace{\{\text{ C, }\{\text{ }\{\text{ the, man}'\text{ }\}\text{, }\{\text{ PRES}'\text{, }\{\text{ }v\text{, }\{\text{ falls, }\{\text{ the, man}'\text{ }\}\ \}\ \}\ \}\ \}\ \}}_{\text{CP}}\}$.

g. Transfer(CP, CP).
$W_7 = \{\cancel{\text{CP}}\}$.

(22) a. Select(*man*).
$W_1 = \{\text{man}\}$.

b. Select(PRES).
$W_2 = \{\text{man}, \text{PRES}\}$.

c. Merge(*man*, [iC:nom]). [iC:nom] is in the SYN of PRES.
$W_3 = \{\underbrace{\{\text{ man, [iC:nom] }\}}_{\text{N}}, \text{PRES}\}$.

d. Merge(*the*, N).
$W_4 = \{\underbrace{\{\text{ the, \{ man, [iC:nom] } \}}_{\text{DP}}, \text{PRES}\}$.

e. Merge(*falls*, DP).
$W_5 = \{\underbrace{\{\text{ falls, \{ the, \{ man, [iC:nom] } \} }\}}_{\text{VP}}, \text{PRES}\}$.

f. Merge(v, VP).
$W_6 = \{\underbrace{\{\text{ } v\text{, \{ falls, \{ the, \{ man, [iC:nom] } \} \} }\}}_{v\text{P}}, \text{PRES}\}$.

g. Merge(PRES, vP).
$W_7 = \{\underbrace{\{\text{ PRES, \{ } v\text{, \{ falls, \{ the, \{ man, [iC:nom] } \} \} \} }\}}_{\text{TP}_1}\}$.

h. Merge(TP_1, DP).
$W_8 = \{\underbrace{\{\text{ \{ the, \{ man, [iC:nom] } \}, \{ PRES, \{ } v\text{, \{ falls, \{ the, \{ man, [iC:nom] } \} \} \} \} \} }\}}_{\text{TP}_2}\}$.

i. Merge(TP_2, [iπ:3]). [iπ:3] is in the SYN of *man*.
$W_9 = \{\underbrace{\{\text{ [i}\pi\text{:3], \{ DP, \{ PRES, } v\text{P } \} \} }\}}_{\text{TP}_3}\}$.

j. Merge(TP_3, [i#:sg]). [i#:sg] is in the SYN of *man*.
$W_{10} = \{\underbrace{\{\text{ [i\#:sg], \{ [i}\pi\text{:3], \{ DP, \{ PRES, } v\text{P } \} \} \} }\}}_{\text{TP}_4}\}$.

k. Merge(C, TP_4).
$W_{11} = \{\underbrace{\{\text{ C, \{ [i\#:sg], \{ [i}\pi\text{:3], \{ DP, \{ PRES, } v\text{P } \} \} \} \} }\}}_{\text{CP}}\}$.

l. Transfer(CP, CP).
$W_{12} = \{\text{CP}\}$.

Does a neural model understand the *de re* / *de dicto* distinction?

Gaurav Kamath[1,*]
gaurav.kamath@mail.mcgill.ca

Laurestine Bradford[1,2,*]
laurestine.bradford@mail.mcgill.ca

[1] McGill University and Mila
[2] Centre for Research on Brain, Language and Music

Abstract

Neural network language models (NNLMs) are often casually said to "understand" language, but what linguistic structures do they really learn? We pose this question in the context of *de re* / *de dicto ambiguities*. Nouns and determiner phrases in intensional contexts, such as belief, desire, and modality, are subject to referential ambiguities. The phrase "Lilo believes an alien is on the loose," for example, has two interpretations: one (*de re*) in which she believes a specific entity which happens to be an alien is on the loose, and another (*de dicto*) in which she believes some unspecified alien is on the loose. In this paper we confront an NNLM with contexts producing *de re* / *de dicto* ambiguities. We use coreference resolution to investigate which interpretive possibilities the model captures. We find that while RoBERTa is sensitive to the fact that intensional predicates and indefinite determiners each change coreference possibilities, it does not grasp how the two interact with each other, and hence misses a deeper level of semantic structure. This inquiry is novel in its cross-disciplinary approach to philosophy, semantics and NLP, bringing formal semantic insight to an active research area testing the nature of NNLMs' linguistic "understanding."

1 Introduction

Modern neural net language models (NNLMs) are often publicized as "understanding" language, which can belie a lack of knowledge about the nature of the linguistic structures they truly capture (Bender and Koller, 2020). Consequently, there has been much interest in probing NNLMs' sensitivity to theoretical linguistic structures, an area which Baroni (2021) calls *linguistically-oriented deep net analysis* (LODNA). Such analysis often uses psycholinguistic methods to give NNLMs acceptability tasks similar to those one would give to a human (Warstadt et al., 2019). Existing work has primarily measured NNLMs' ability to capture syntactic structures (Bacon, 2020; Linzen and Baroni, 2021; Warstadt et al., 2019), though a few semantic phenomena, such as the causative-inchoative alternation, have also been investigated (Warstadt et al., 2019).

Fine-grained semantic distinctions present unique difficulties for LODNA. It can be challenging to pose the right problems to test NNLM knowledge of subtle meaning distinctions; for example, see (Tsiolis, 2020)'s discussion in the context of quantifier scope ambiguity. Nonetheless, fine-grained semantic distinctions are crucial to modern theories of semantic structure, and it is therefore important to find out how well NNLMs "understand" them. One such subtle meaning difference lies in the *de re* and *de dicto* interpretations of noun phrases in intensional contexts.

The *de re* / *de dicto* distinction, made notable by Quine (1956) among others, refers to two distinct kinds of interpretations of noun phrases that arise from intensional contexts in natural language. Such contexts include belief, desire, and modality. The statement "Lilo believes an alien is on the loose," for example, has two interpretations. Under one interpretation (*de re*), Lilo believes a specific entity that just so happens to be an alien (say, Stitch) is on the loose. Lilo herself (as is the case in *Lilo and Stitch* (Sanders and DeBlois, 2002)) need not know that Stitch is an alien for the statement to be true. Under the other interpretation (*de dicto*) Lilo believes that some unspecified alien, whatever it may be, is on the loose. Unlike the *de re* interpretation, no alien needs to actually exist for the statement to be true under this interpretation.

De re / *de dicto* ambiguities have traditionally been treated in the philosophy and semantics literature as scope ambiguities, where each interpretation arises out of a modal or intensional operator outscoping, or being outscoped by, another

[*] Equal contribution.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 69-84.
Amherst, Massachusetts, June 15-17, 2023

quantifier (see (Keshet and Schwarz, 2019) for an overview). For example:

De re: $\exists x[\text{alien}_{w_0}(x) \wedge \forall w'$ $[\text{BEL}_{w_0}(\text{Lilo},w') \Rightarrow \text{on-the-loose}_{w'}(x)]]$

De dicto: $\forall w'[\text{BEL}_{w_0}(\text{Lilo},w') \Rightarrow$ $\exists x[\text{alien}_{w'}(x) \wedge \text{on-the-loose}_{w'}(x)]]$[1]

NNLMs, however, lack any similar formal system of representation, since all meaning representation is contained within numerical embeddings and weights. This provides further theoretical motivation to investigate whether NNLMs are capable of discerning *de re* / *de dicto* ambiguities, and whether they show any bias towards either interpretation. If NNLMs are capable of making these distinctions, it would suggest not only that they are capable of mimicking human-like fine-grained semantic distinctions, but also that numerical vectors are rich enough to capture deep formal structure. We thus believe that the capacity of NNLMs to discern *de re* / *de dicto* ambiguities has strong implications for both semantics and NLP.

Therefore, we investigate whether current powerful language models can interpret NPs in intensional contexts in both *de re* and *de dicto* senses. We will do so by framing the problem as one of coreference resolution.

2 Related Work

As NNLMs have become increasingly successful at a range of natural language tasks in recent years, there has been much discussion of the capacity of such models to "understand" language. While this use of the term is misleading (Bender and Koller, 2020), it has spurred research into the ability of NNLMs to pick up on theoretical, often complex linguistic structures.

Most of this LODNA work has focused on syntactic structures. For overviews of such work, see (Baroni, 2021; Bender and Koller, 2020; Linzen and Baroni, 2021). The present paper differs from this body of work, however, in that we address a semantic, rather than a syntactic, phenomenon.

Although not as much, there has also been work in LODNA on semantics. For example, some progress has been made in measuring the degree to which NNLMs encode compositionality (Ettinger et al., 2018; Shwartz and Dagan, 2019; Jawahar et al., 2019; Yu and Ettinger, 2020, 2021; Bogin et al., 2022) and systematicity (Lake and Baroni, 2018; Goodwin et al., 2020; Kim and Linzen, 2020). Researchers have also studied the capacity of NNLMs to capture more specific, fine-grained semantic phenomena, including monotonicity (Yanaka et al., 2019), the causative-inchoative alternation (Warstadt et al., 2019), negation (Ettinger et al., 2018; Ettinger, 2020; Kim et al., 2019; Richardson et al., 2020), and quantification (Kim et al., 2019; Richardson et al., 2020).

Natural language understanding (NLU) benchmarks also have the opportunity to test models' grasp of theoretical semantic structures. Most large collections of NLU benchmarks focus on performance of specific tasks (such as sentiment analysis and question answering) rather than abstract linguistic knowledge (Liang et al., 2020; Ruder et al., 2021; Dumitrescu et al., 2021; Ham et al., 2020; Khashabi et al., 2020; Park et al., 2021; Rybak et al., 2020; Seelawi et al., 2021; Wilie et al., 2020; Yao et al., 2021). Indeed, Bowman and Dahl (2021) have argued that targeting specific linguistic knowledge can hinder performance of NNLMs on NLP tasks.

Nevertheless, some NLU benchmarks overlap with LODNA in addressing certain theoretical semantic structures. In particular, the benchmarks discussed in (Xia and Van Durme, 2021) all assess models' semantically-informed coreference resolution capability, as do the collection of benchmarks following the Winograd Schema (Levesque et al., 2012; Kocijan et al., 2020), which includes some large benchmark sets like those mentioned above (Wang et al., 2019a,b; Xu et al., 2020; Shavrina et al., 2020). A benchmark nearer to the spirit of LODNA is proposed in (Yanaka et al., 2021). This paper directly relates generation of NNLM test cases to theoretical semantic structures. The authors use such structures to create tests for NNLMs' compositional generalization of logical operators, modifiers, and embedded clauses. Finally, in the class of NLU benchmarks, the work of (Ribeiro et al., 2021) is nearest to our own investigation. Here, the author proposes templates that can be filled in to create probes of NNLMs' capability with a variety of structures. These structures include antonymy, temporal ordering, negation, and coreference. Note that none of the previous work

[1] While other equivalent formulations of the logical forms of such sentences are present in the literature, we choose to adopt the same notation as (Zhang and Davidson, 2021), on account of its conciseness and simplicity.

assesses modality or intensionality. In the present work, we employ a template-like scheme for generating test cases that assess NNLMs' behaviour in intensional contexts.

We focus on the *de re* / *de dicto* distinction. Since being highlighted in recent times by (Quine, 1956), *de re* / *de dicto* ambiguities have been the subject of extensive work in philosophy and semantics. For an overview, see (Keshet and Schwarz, 2019). Most of this work focuses on of how to formally represent intensional contexts (Fodor, 1970; Tichý, 1971; Montague, 1973; Lewis, 1979; Von Fintel and Heim, 2011); specific points of focus include scope (Keshet, 2008, 2010), (Elliott, 2022), modality (Plantinga, 1969; Fine, 1978), and even tense (Ogihara, 1996; Kauf and Zeijlstra, 2018). For all this work on the theory of *de re* / *de dicto* ambiguities, however, there is a dearth of experimental work on the distinction. The work reported in (Zhang and Davidson, 2021) therefore stands out for its quantitative experimental approach. The authors conduct an study directly measuring whether English speakers demonstrate any preference towards *de re* or *de dicto* readings. Their results suggest that speakers accept *de dicto* interpretations more robustly than *de re* interpretations.

To our knowledge, there has been no similar attempt to situate *de re* / *de dicto* ambiguities in the context of NNLMs. Williamson et al. (2021) present an amendment to Abstract Meaning Representation (AMR), a graphical meaning representation language, which allows it to encode *de re* / *de dicto* ambiguities as scope ambiguities. This marks perhaps the closest recent work on these ambiguities in a NLP context. AMR, however, is an artificial meaning representational language, and therefore of a different type than the meaning representation of an NNLM. Our work directly looks for *de re* / *de dicto* ambiguities in NNLMs' behaviour.

3 Model

In all experiments, we use a version of the RoBERTa (Liu et al., 2019) masked language model already fine-tuned for the SuperGLUE Winograd Schema Challenge task (Levesque et al., 2012; Wang et al., 2019a). This is because: (i) our method of distinguishing *de re* from *de dicto* interpretations centers on recognizing coreference, which this model does well at, scoring 89% on the SuperGLUE WSC task (while for comparison, OpenAI's few-shot GPT-3 scores 80.1%) (Wang et al.); and (ii) this model proved most straightforward to access and work with. We directly access and work with this model using Meta AI's fairseq library (Ott et al., 2019).

4 Dataset and evaluation metric

4.1 Dataset

We generate a dataset of test sentences that consist of a matrix subject, an intensional verb with sentential complement, an embedded subject, and an embedded intransitive verb. The matrix subject is always *John* or *Mary*, and the embedded subject is always a noun phrase. All of the test cases have either the form in Figure 1a, as in the example *John believes that a dentist is singing*, or the form in Figure 1b, as in the example *John wants a dentist to be singing*. The choice between these structures simply depends on whether the matrix verb requires a finite or non-finite tense in its complement.

We simultaneously generate a dataset of sentences which are similar to the above, but with a perceptual verb instead of an intensional verb. These therefore have the form in Figure 1c, as in the example *John sees a dentist singing*. Note that perceptual verbs have been analyzed by a few in the literature as also being intensional (e.g. Bourget, 2017); for sentences with perceptual verbs, we therefore have the perceptual verbs take direct objects as their arguments (as in *John sees a dentist singing*), rather than clauses (as in *John sees that a dentist is singing*), so as to minimize the possibility of intensional interpretations of the perceptual verbs.

Sentence templates are generated from the schemata in Figure 1 with every possible combination of: *John* or *Mary* in the matrix subject, a verb from the list in Appendix A.3 in the matrix verb, a noun from the list in Appendix A.1 in the embedded subject, and a verb from the list in Appendix A.2 in the embedded verb.

In addition to manipulating whether the matrix verb is intensional, we manipulate the determiner of the embedded subject. We generate alternations between the indefinite determiner 'a'/'an', as in *Mary believes that a dentist is smiling*, and the deictic determiner 'that', as in *Mary believes that that dentist is smiling*. The indefinite 'a'/'an' should give rise to a *de re* / *de dicto* ambiguity. The deictic 'that' should, in theory, only allow for a *de re* interpretation, since it must refer to an entity already present in the world of discourse.

[MatrixSubject]	[MatrixVerb] *that*	[EmbeddedSubject] *is*	[EmbeddedVerb]
John	*believes*	*an editor*	*walking*
Mary	*accepts*	*a dentist*	*singing*
	deduces	*a baker*	*shouting*
	...	...	...

(a) Intensional sentences with finite-tensed complements.

[MatrixSubject]	[MatrixVerb]	[EmbeddedSubject] *to be*	[EmbeddedVerb]
John	*wants*	*an editor*	*walking*
Mary	*wishes for*	*a dentist*	*singing*
	requires	*a baker*	*shouting*
	...	...	...

(b) Intensional sentences with non-finite-tensed complements.

[MatrixSubject]	[MatrixVerb]	[EmbeddedSubject]	[EmbeddedVerb]
John	*sees*	*an editor*	*walking*
Mary	*observes*	*a dentist*	*singing*
	hears	*a baker*	*shouting*
	...	...	...

(c) Perceptual sentences.

Figure 1: Schemata for generating test data

We handpick 48 matrix verbs (36 intensional and 12 perceptual), randomly select 60 embedded nouns from a handpicked list of 204, and randomly select 30 embedded verbs from a handpicked list of 51[2]. The resultant dataset contains a total of 345,600 unique sentences with the configurations shown in Figure 1 (although the total size of dataset is larger, for reasons explained in the following section). 259,200 of these are sentences with intensional verbs, and the remaining 86,400 are sentences with perceptual verbs.

4.2 Evaluation

The availability of the embedded NP as an anaphoric antecedent depends on whether it is interpreted *de re* or *de dicto*. Consequently, for each generated sentence, we post-pend three different fixed sentences: (i) *I met [pronoun]*, (ii) *I greeted [pronoun]*, and (iii) *I liked [pronoun]*[3]. We then use a tweaked version of the WSC-finetuned RoBERTa model's in-built `disambiguate_pronoun` function to obtain the scores the model assigns at the *[pronoun]* position to each possible coreferent (i.e. the main subject or the embedded subject)[4].

Under the *de dicto* reading, the embedded NP should not be able to corefer with a subsequent phrase, as under this reading it is interpreted solely within the intensional context. By contrast, under the *de re* reading, the embedded NP should be able to corefer with a subsequent phrase, as under this reading it is interpreted outside the intensional context.

In intuitive terms, using the example *Mary believes that a lawyer is shouting*, under the *de dicto* interpretation, the lawyer is only specified in Mary's beliefs, rather than the speaker's world of reference. But the subsequent post-pended sentence is evaluated with respect to the speaker's world of reference, and not Mary's beliefs. So, the pronoun token in the post-pended sentence should not be able to refer to the embedded NP. Under a *de re* interpretation, however, the lawyer is specified in the speaker's world of reference. So it remains accessible for coreference in the post-pended sentence.

Therefore, we should be able to assess the perfor-

[2]We randomly select subsets of these lists, instead of using the entire handpicked lists, due to concerns of dataset size and excessive compute requirements with little obvious *a priori* benefit of using the complete lists.

[3]This triples the final size of our dataset, bringing it to 1,036,800.

[4]In this process, the model doesn't actually make use of the token in the position it predicts for. We therefore use the *[pronoun]* token as a placeholder for what is in effect a masked position, as using RoBERTa's actual <*mask*> token led to issues with the code.

mance of the masked language model at detecting the *de re* / *de dicto* ambiguity by comparing the scores it assigns to the matrix or the embedded subject at the pronoun position. For example, in *Mary believes that a dentist is singing. I met [pronoun]*, we compare the scores assigned to the possible coreferents *Mary* and *a dentist* at the pronoun position[5]. We use three separate post-pended sentences to try to ensure that the effects we see are not the result of any one specific verb in the follow-up sentence.

Scores assigned to the matrix subject should be higher for test sentences where the matrix verb is intensional and the embedded subject has an 'a'/'an' determiner. These are the contexts that give rise to the possible *de dicto* interpretation which would exclude the embedded subject from coreference. By contrast, the relative scores for the matrix and embedded subject should be closer to equal in cases that only admit a *de re* interpretation. This includes all cases with a 'that' determiner or where the matrix verb is perceptual (i.e. non-intensional).

5 Results and Discussion

5.1 Results

To quantify the model's coreference choice at the pronoun position, we study the difference between the score assigned to the matrix subject (e.g. *John*) and that assigned to the embedded subject (e.g. *an actor*); we call this difference *matrix subject bias*. Figure 2 shows the empirical effect of matrix verb type and determiner type on matrix subject bias. We see an overall increase in matrix subject bias in intensional contexts and in contexts where the embedded subject has the determiner 'a' or 'an'. The difference between intensional and perceptual contexts is slightly smaller when the embedded subject has determiner 'a' or 'an'.

In order to study the effects of interest while marginalizing over other manipulations and over random variability, we fit a linear mixed-effects model with formula below (random effects specified in brackets).

$$
\begin{aligned}
&\text{Matrix Subject Bias} \sim \\
&1 + \text{Determiner} * \text{Matrix Verb Type} \\
&\qquad + \text{Followup Verb} + \text{Matrix Subject} \\
&+ (1 + \text{Determiner} + \text{Matrix Subject} \\
&\qquad + \text{Followup Verb} \mid \text{Matrix Verb}) \\
&+ (1 + \text{Determiner} * \text{Matrix Verb Type} \\
&\qquad + \text{Followup Verb} + \text{Matrix Subject} \\
&\qquad \mid \text{Embedded Verb}) \\
&+ (1 + \text{Determiner} * \text{Matrix Verb Type} \\
&\qquad + \text{Followup Verb} + \text{Matrix Subject} \\
&\qquad \mid \text{Embedded Subject})
\end{aligned}
$$

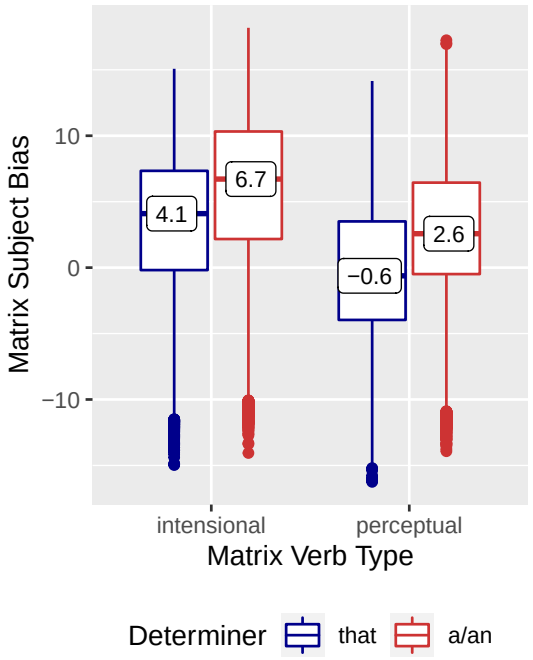


Figure 2: Boxplot with whiskers to 1.5IQR showing distribution of matrix subject bias by determiner and matrix verb type.

The full results are reported in Tables 1 and 2. The model confirms the overall trend in Figure 2. Averaged across all conditions, there is a bias towards matrix subjects of 3.27 points (df=71.91, t=6.96, p<0.001). Sentences with perceptual matrix verbs show 2.58 points lower matrix subject bias than those with intensional matrix verbs (df=78.16, t=-5.03, p<0.001), and sentences with with determiner 'a'/'an' show 2.89 points higher matrix subject bias than those with determiner 'that' (df=92.78, t=11.92, p<0.001). The effect of verb type is smaller in indefinite ('a'/'an') determiner contexts than deictic ('that') contexts by 0.52 points, but this is not statistically significant (df=72.83,t=1.44,p=0.152).

There is considerable variability in both effects according to embedded verb and embedded subject, and variability in the determiner effect according to matrix verb, embedded verb, and embedded subject

[5]The implementation of coreference resolution in the model we use is such that a span such as *a dentist* is not penalized simply for being longer than a single token like *Mary*.

Coefficient	$\hat{\beta}$	**SE**$(\hat{\beta})$	**df**	**t**	**p**
Intercept	3.27	0.47	71.91	6.96	< 0.001
Determiner = 'a/an'	2.89	0.24	92.78	11.92	< 0.001
Matrix Verb Type = 'perceptual'	-2.58	0.51	78.16	-5.03	< 0.001
Matrix Subject = 'Mary'	-1.27	0.17	89.18	-7.65	< 0.001
Followup Verb = 'liked' (vs. 'greeted')	-0.25	0.26	102.91	-0.97	0.333
Followup Verb = 'met' (vs. 0.5('liked'+'greeted'))	-1.12	0.13	96.10	-8.93	< 0.001
Interaction Determiner:Matrix Verb Type	0.52	0.36	72.83	1.44	0.152
Marginal $R^2 = 0.21$, Conditional $R^2 = 0.65$, $n = 1036800$,					
Groups: Matrix Verb (48); Embedded Verb (30); Embedded Subject (60)					

Table 1: A regression table showing fixed effects, goodness of fit, and test statistics for the linear mixed-effects model in Section 5.1. Degrees of freedom and p-values estimated using the Satterthwaite approximation. Predictor levels were coded as ± 0.5, except Followup Verb coded with Helmert contrasts.

Group	**Term**	**Variance**	**SD**
Matx. Verb	Intercept	1.13	1.49
	Determiner	0.89	0.94
	Matx. Subj	0.05	0.22
	Foll. Verb Cont.1	1.12	1.05
	Foll. Verb Cont.2	0.23	0.48
Emb. Verb	Intercept	3.92	1.98
	Determiner	0.76	0.87
	Matx. Verb Type	2.02	1.42
	Matx. Subj	0.17	0.42
	Foll. Verb Cont.1	0.84	0.92
	Foll. Verb Cont.2	0.22	0.47
	Det.:Matx. Type	0.80	0.90
Emb. Subj	Intercept	1.92	1.39
	Determiner	0.50	0.71
	Matx. Verb Type	0.79	0.89
	Matx. Subj	1.25	1.12
	Foll. Verb Cont.1	0.88	0.93
	Foll. Verb Cont.2	0.21	0.46
	Det.:Matx. Type	0.38	0.62
Residual		10.09	3.18

Table 2: A table showing fitted random effects of the model specified in Section 5.1, as well as residual variance.

(Table 2). Nonetheless, the overall trend is clear.

See Appendix B for an overview of additional trends which do not bear on the main research question.

5.2 Discussion

From these results, it is clear that both verb type (intensional or non-intensional) and determiner type (indefinite or deictic) have statistically significant effects on the relative scores the language model assigns to different possible anaphoric referents.

Intensional verbs yield higher matrix subject bias than non-intensional, perceptual verbs, when all other variables are held constant. This is in line with our predictions, as intensional verbs allow for *de dicto* readings that block the embedded subject from coreference.

In addition, indefinite determiners yield higher matrix subject bias than deictic determiners. This is also in line with our predictions, as indefinite determiners are more amenable to *de dicto* readings that block the embedded subject from coreference. However, the interaction between these two factors is not statistically significant. This goes against our predictions, as deictic determiners should bias the reader toward *de re* readings no matter what, so the matrix verb effect should diminish when the determiner is 'that'.

These results are positive evidence that neural language models can be sensitive to the effect of intensional predicates on *de re* / *de dicto* ambiguities, and therefore to intensionality more broadly. However, the lack of interaction suggests that there is something deeper that RoBERTa misses. It captures the effects of verb intensionality and deictic determiners; however, it does not capture the correct result of combining the two. By contrast, a formal-theoretical model of intensional verbs' and of determiners' meanings would lead naturally to the correct inference that deictic determiners facilitate *de re* readings regardless of matrix verb.

Some other results are also worth mentioning, shown in more detail in Appendix B. As seen in Table 1 and Figure 4b, the matrix subject bias is very similar when the followup verb is *liked* or *greeted*, but lower in a statistically significant way when

it is *met*. The reason for this effect is not known. Whether the matrix subject is *Mary* or *John* has a statistically significant effect on matrix subject bias; holding other variables constant, setting the matrix subject to *Mary* instead of *John* yields a lower matrix subject bias. Given the propensity for large language models to be gender-biased in various ways (Lu et al., 2020; Vig et al., 2020; Charlesworth et al., 2021), this is perhaps not surprising.

6 Conclusion

In this paper, we investigate the capacity of a neural language model, a version of RoBERTa fine-tuned for coreference resolution, to identify *de re* / *de dicto* ambiguities that arise in intensional contexts. We find evidence suggesting that such models are indeed sensitive to the ambiguity-generating effects of intensional predicates and the ambiguity-resolution effects of deictic determiners, but find no evidence that this sensitivity extends to the interaction between intensional predicates and embedded determiners.

Our approach is also subject to some limitations that invite further research. Our range of test data is tightly constrained in its syntactic and broad semantic structure. This is deliberate, as we hoped to isolate the semantic effects of intensional predicates and determiners from the confounding factors of syntactic form and broader semantic context. However, the downside of this approach is that our findings may not generalize across more varied forms of language. Similarly, our choice of perceptual verbs as the counterpart to intensional verbs was the result of their shared syntactic properties, which allowed for substitution while holding all other variables (including sentence structure) virtually unchanged. One possibility, however, is that the effects we find between intensional and perceptual verbs are dependent on the latter's being specifically perceptual verbs, and do not represent a difference between intensional and non-intensional verbs more generally. Finally, in this paper, we work with only one model. Other models with different architecture or pretraining may have produced different results.

Clearly, a broader study of the capacity of neural models to capture intensional effects such as *de re* / *de dicto* ambiguities requires a wider set of data and experimental setups. We hope that this inquiry spurs further research to that end.

7 Code

Code and data for this project are available at https://github.com/laurestine/nnlm-de-re-de-dicto.

8 Acknowledgements

The authors would like to thank Siva Reddy for his guidance, as well as Chris Potts and the anonymous reviewers for their feedback on earlier versions of this work. The CRBLM is funded by the Government of Quebec via the Fonds de Recherche Nature et Technologies and Société et Culture.

References

Geoffrey I. Bacon. 2020. *Evaluating linguistic knowledge in neural networks*. Ph.D. thesis, UC Berkeley.

Marco Baroni. 2021. On the proper role of linguistically-oriented deep net analysis in linguistic theorizing.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Ben Bogin, Shivanshu Gupta, and Jonathan Berant. 2022. Unobserved local structures make compositional generalization hard. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2731–2747, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Bourget. 2017. Intensional perceptual ascriptions. *Erkenntnis volume*, 82.

Samuel R. Bowman and George Dahl. 2021. What will it take to fix benchmarking in natural language understanding? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4843–4855, Online. Association for Computational Linguistics.

Tessa E. S. Charlesworth, Victor Yang, Thomas C. Mann, Benedek Kurdi, and Mahzarin R. Banaji. 2021. Gender stereotypes in natural language: Word embeddings show robust consistency across child and adult language corpora of more than 65 million words. *Psychological Science*, 32(2):218–240. PMID: 33400629.

Stefan Daniel Dumitrescu, Petru Rebeja, Beata Lorincz, Mihaela Gaman, Andrei Avram, Mihai Ilie, Andrei Pruteanu, Adriana Stan, Lorena Rosia, Cristina Iacobescu, Luciana Morogan, George Dima, Gabriel

Marchidan, Traian Rebedea, Madalina Chitez, Dani Yogatama, Sebastian Ruder, Radu Tudor Ionescu, Razvan Pascanu, and Viorica Patraucean. 2021. LiRo: Benchmark and leaderboard for Romanian language tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Patrick D Elliott. 2022. A flexible scope theory of intensionality. *Linguistics and Philosophy*, 46:333–378.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing composition in sentence vector representations. In *COLING*.

Kit Fine. 1978. Model theory for modal logic. part i – the de re/de dicto distinction. *Journal of Philosophical Logic*, 7(1):125–156.

Janet Dean Fodor. 1970. *The Linguistic Description of Opaque Contexts*. Ph.D. thesis, MIT.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969, Online. Association for Computational Linguistics.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020. KorNLI and KorSTS: New benchmark datasets for Korean natural language understanding. *CoRR*, abs/2004.03289.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Carina Kauf and Hedde Zeijlstra. 2018. Towards a new explanation of sequence of tense. In *Semantics and Linguistic Theory*, volume 28, pages 59–77.

Ezra Keshet. 2008. *Good intensions: paving two roads to a theory of the de re / de dicto distinction*. Ph.D. thesis, MIT.

Ezra Keshet. 2010. Split intensionality: A new scope theory of de re and de dicto. *Linguistics and Philosophy*, 33(4):251–283.

Ezra Keshet and Florian Schwarz. 2019. De re/de dicto. *The Oxford handbook of reference*, pages 167–202.

Daniel Khashabi, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, Mozhdeh Gheini, Arman Kabiri, Rabeeh Karimi Mahabadi, Omid Memarrast, Ahmadreza Mosallanezhad, Erfan Noury, Shahab Raji, Mohammad Sadegh Rasooli, Sepideh Sadeghi, Erfan Sadeqi Azer, Niloofar Safi Samghabadi, Mahsa Shafaei, Saber Sheybani, Ali Tazarv, and Yadollah Yaghoobzadeh. 2020. ParsiNLU: A suite of language understanding challenges for Persian. *CoRR*, abs/2012.06154.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Vid Kocijan, Thomas Lukasiewicz, Ernest Davis, Gary Marcus, and Leora Morgenstern. 2020. A review of Winograd schema challenge datasets and approaches. *CoRR*, abs/2004.13831.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012*, Proceedings of the International Conference on Knowledge Representation and Reasoning, pages 552–561. Institute of Electrical and Electronics Engineers Inc. 13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012 ; Conference date: 10-06-2012 Through 14-06-2012.

David Lewis. 1979. Attitudes de dicto and de se. *Philosophical Review*, 88(4):513–543.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7(1):195–212.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. *Gender Bias in Neural Natural Language Processing*, pages 189–202. Springer International Publishing, Cham.

Richard Montague. 1973. The proper treatment of quantification in ordinary english. In Patrick Suppes, Julius Moravcsik, and Jaakko Hintikka, editors, *Approaches to Natural Language*, pages 221–242. Dordrecht.

Toshiyuki Ogihara. 1996. *Tense, attitudes, and scope*, volume 58 of *Studies in Linguistics and Philosophy*. Springer Science & Business Media.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won-Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Tae Hwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Eunjeong Lucy Park, Alice Oh, Jung-Woo Ha, and Kyunghyun Cho. 2021. KLUE: Korean language understanding evaluation. *CoRR*, abs/2105.09680.

Alvin Plantinga. 1969. De re et de dicto. *Noûs*, 3(3):235–258.

Willard Quine. 1956. Quantifiers and propositional attitudes. *Journal of Philosophy*, 53:177–187.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2021. Beyond accuracy: Behavioral testing of NLP models with checklist (extended abstract). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4824–4828. International Joint Conferences on Artificial Intelligence Organization. Sister Conferences Best Papers.

Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8713–8721.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. 2020. KLEJ: comprehensive benchmark for Polish language understanding. *CoRR*, abs/2005.00630.

Chris Sanders and Dean DeBlois. 2002. *Lilo & Stitch*. Walt Disney Pictures.

Haitham Seelawi, Ibraheem Tuffaha, Mahmoud Gzawi, Wael Farhan, Bashar Talafha, Riham Badawi, Zyad Sober, Oday Al-Dweik, Abed Alhakim Freihat, and Hussein Al-Natsheh. 2021. ALUE: Arabic language understanding evaluation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 173–184, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Tatiana Shavrina, Alena Fenogenova, Anton A. Emelyanov, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. RussianSuperGLUE: A Russian language understanding evaluation benchmark. *CoRR*, abs/2010.15925.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Pavel Tichý. 1971. An approach to intensional analysis. *Noûs*, 5(3):273–297.

Konstantinos Christopher Tsiolis. 2020. Quantifier scope disambiguation.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Kai Von Fintel and Irene Heim. 2011. Intensional semantics. *Unpublished Lecture Notes*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE leaderboard. Available at `https://super.gluebenchmark.com/leaderboard` (2022/04/18).

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. *SuperGLUE: A Stickier Benchmark for General-Purpose Language*

Understanding Systems. Curran Associates Inc., Red Hook, NY, USA.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. *CoRR*, abs/2009.05387.

Gregor Williamson, Patrick Elliott, and Yuxin Ji. 2021. Intensionalizing Abstract Meaning Representations: Non-veridicality and scope. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 160–169, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Xia and Benjamin Van Durme. 2021. Moving on from OntoNotes: Coreference resolution model transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5241–5256, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liang Xu, Xuanwei Zhang, Lu Li, Hai Hu, Chenjie Cao, Weitang Liu, Junyi Li, Yudong Li, Kai Sun, Yechen Xu, Yiming Cui, Cong Yu, Qianqian Dong, Yin Tian, Dian Yu, Bo Shi, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. *CoRR*, abs/2004.05986.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40, Florence, Italy. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. SyGNS: A systematic generalization testbed based on natural language semantics. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 103–119, Online. Association for Computational Linguistics.

Yuan Yao, Qingxiu Dong, Jian Guan, Boxi Cao, Zhengyan Zhang, Chaojun Xiao, Xiaozhi Wang, Fanchao Qi, Junwei Bao, Jinran Nie, Zheni Zeng, Yuxian Gu, Kun Zhou, Xuancheng Huang, Wenhao Li, Shuhuai Ren, Jinliang Lu, Chengqiang Xu, Huadong Wang, Guoyang Zeng, Zile Zhou, Jiajun Zhang, Juanzi Li, Minlie Huang, Rui Yan, Xiaodong He, Xiaojun Wan, Xin Zhao, Xu Sun, Yang Liu, Zhiyuan Liu, Xianpei Han, Erhong Yang, Zhifang Sui, and Maosong Sun. 2021. CUGE: A Chinese language understanding and generation evaluation benchmark. *CoRR*, abs/2112.13610.

Lang Yu and Allyson Ettinger. 2021. On the interplay between fine-tuning and composition in transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2279–2293, Online. Association for Computational Linguistics.

Lang-Chi Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *EMNLP*.

Yuhan Zhang and Kathryn Davidson. 2021. *De re* interpretation in belief reports: An experimental investigation. In *Experiments in Linguistic Meaning*, volume 1. Linguistic Society of America.

A Lexical items used in stimuli

A.1 Embedded Subjects

We used the following nouns as embedded subjects, sampled randomly from a list of English nouns denoting professions and types of person:

actor
administrator
ambassador
architect
assistant
baker
bartender
boy
chancellor
clerk
clown
controller
cook
cooper
count
courier
dancer
dealer
dentist
designer
dictator
diver
drummer

economist
editor
emperor
engineer
farmer
girl
governor
guard
guitarist
historian
journalist
king
lady
lawyer
lieutenant
lobbyist
lord
magician
manager
mayor
merchant
model
negotiator
novelist
painter
philosopher
producer
psychiatrist
publisher
queen
rabbi
solicitor
spy
supervisor
treasurer
waiter
woman

A.2 Embedded Verbs

We used the following embedded intransitive verbs, sampled randomly from a list of English intransitive verbs denoting activities.

arriving
coughing
cringing
crying
dying
hiccuping
kneeling
limping
lying
moving
panicking
partying
praying
resting
running
screaming
shouting
sighing
singing
sitting
smiling
smoking
sneezing
standing
sweating
swimming
talking
walking
waving
working

A.3 Matrix Verbs

We used the following intensional matrix verbs, meant to be as wide an array of intensional verbs as possible:

accepts
aims for
anticipates
assumes
believes
concludes
conjectures
deduces
demands for
desires for
doubts
dreads
expects
fears
feels
figures
gathers
guesses
hopes
imagines
intends for
knows
maintains
needs
presumes

reckons
requires
supposes
surmises
suspects
thinks
trusts
understands
wants
wishes for
worries

We used the following perceptual matrix verbs, meant to be as wide an array of perceptual verbs as possible:

catches sight of
detects
glimpses
hears
notices
observes
overhears
perceives
sees
spots
views
watches

B Data distribution details

This appendix contains additional details, not directly relevant to our research questions, about patterns in matrix and embedded subject scores.

Figure 3 shows the raw distribution of matrix and embedded subject scores. Matrix subject scores are generally higher than embedded subject scores.

Figures 4a and 4b show distribution of matrix subject bias for each matrix subject and for each followup. We see that 'met' yields considerably lower matrix subject bias than other followup verbs, while matrix subjects of John are preferred as coreferents more than matrix subjects of Mary.

Figure 5 shows distribution of matrix subject bias for each determiner-syntactic frame pair. We see that the two intensional-verb frames pattern together in the way indicated in the main text: they have higher matrix subject bias than the perceptual-verb frame, and all three frames show higher matrix subject bias with indefinite determiners.

We next computed the raw effect of determiner, the raw effect of intensional matrix verb, and their

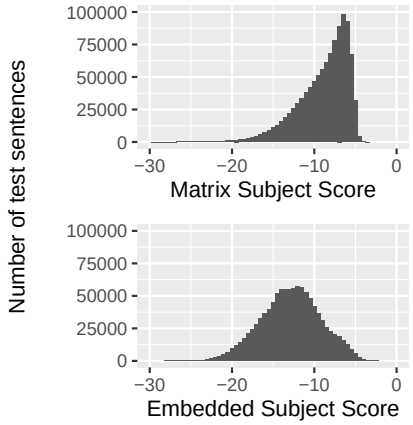
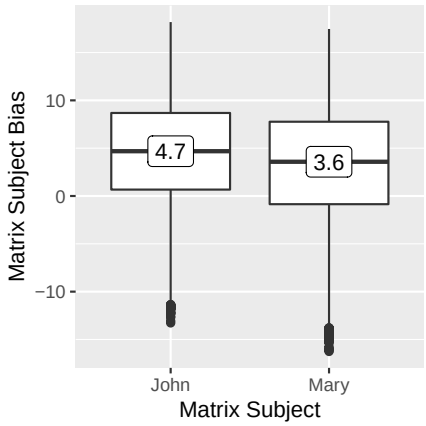
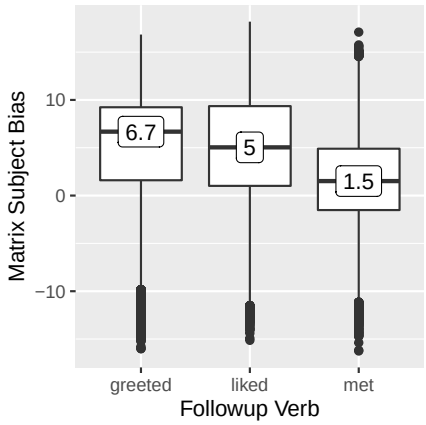


Figure 3: Histograms showing the raw distribution of matrix and embedded subject scores.



(a)



(b)

Figure 4: Boxplot with whiskers to 1.5IQR showing the distribution of matrix subject bias for each matrix subject and for each followup verb.

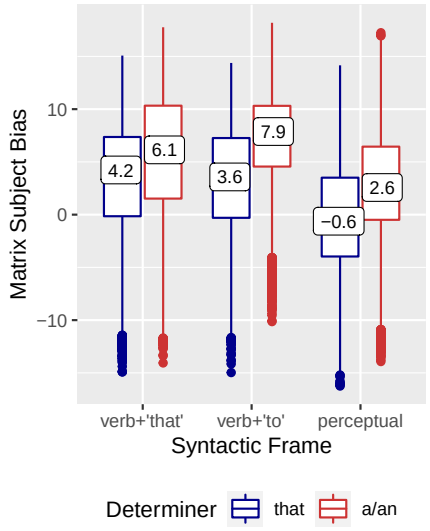


Figure 5: Boxplot with whiskers to 1.5IQR showing the distribution of matrix subject bias by syntactic frame and determiner.

interaction separately for each possible matrix subject, embedded subject, embedded verb, and followup verb. The results are shown in Figure 6. Raw effects are computed as differences of means, and the raw interaction is a difference of differences of means. We see that the overall positive effect of indefinite determiner and intensional matrix verb is a trend across the bulk of data points, and is not merely the result of a few outliers. The lack of interaction between these two effects is also consistent. Figure 7 shows the pattern that test sentence frames with "liked" as a followup verb have a higher effect of determiner than those with other followup verbs, but we see that the effect of an indefinite determiner on matrix subject bias is still positive in general.

Finally, Figures 8, 9, and 10 show variability in matrix subject score and embedded subject score depending on the specific choice of embedded subject (Figure 8), embedded verb (Figure 9), and matrix verb (Figure 10). This variability is quite high, with some lexical items in each case showing almost no matrix subject bias, and others showing quite a lot. Aside from our deliberate manipulation of intensionality, it is unclear what else drives this variability.

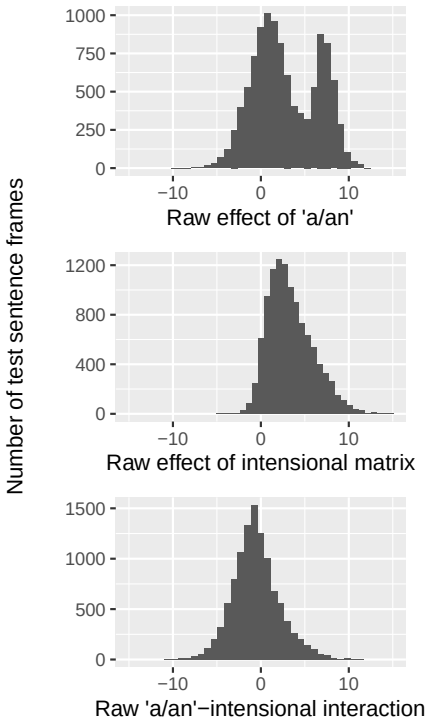


Figure 6: Sentence frames plotted by their raw effect of indefinite determiner (difference in matrix subject bias between instances of that frame with indefinite and deictic determiners), raw effect of intensional matrix verb (difference in mean matrix subject bias between instances of that frame with an intensional and perceptual matrix verb), and raw interaction of these two effects (difference-of-differences between the aforementioned subgroups).

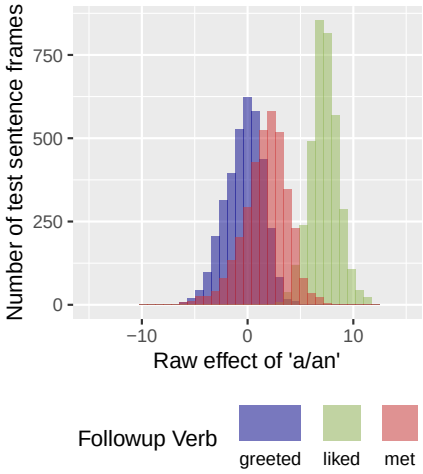


Figure 7: Sentence frames plotted by their raw effect of indefinite determiner, colored by followup verb.

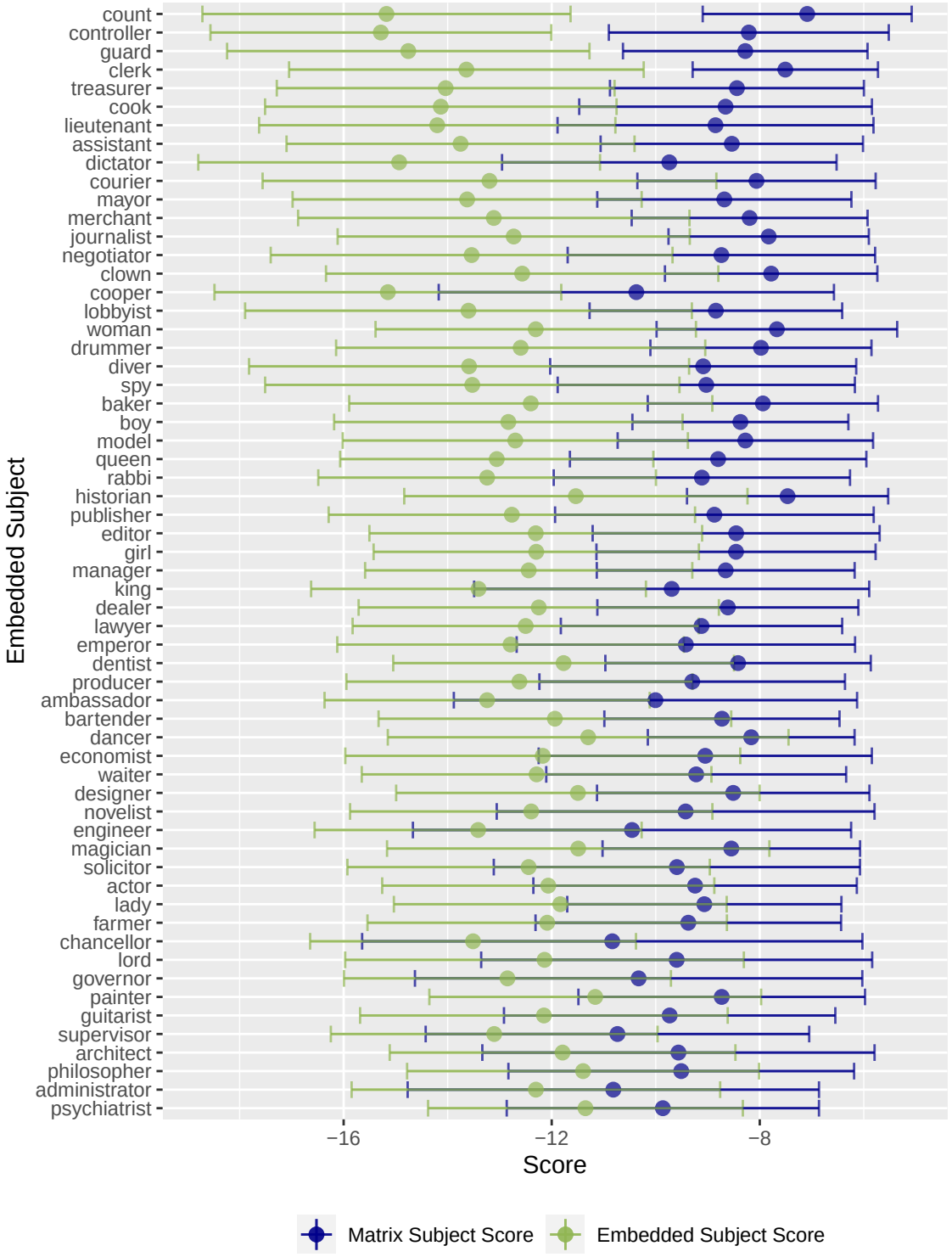


Figure 8: Error bar plot showing mean matrix subject score and embedded subject score for stimuli with each embedded subject. Rows are ordered by matrix subject bias. Error bars show standard deviation.

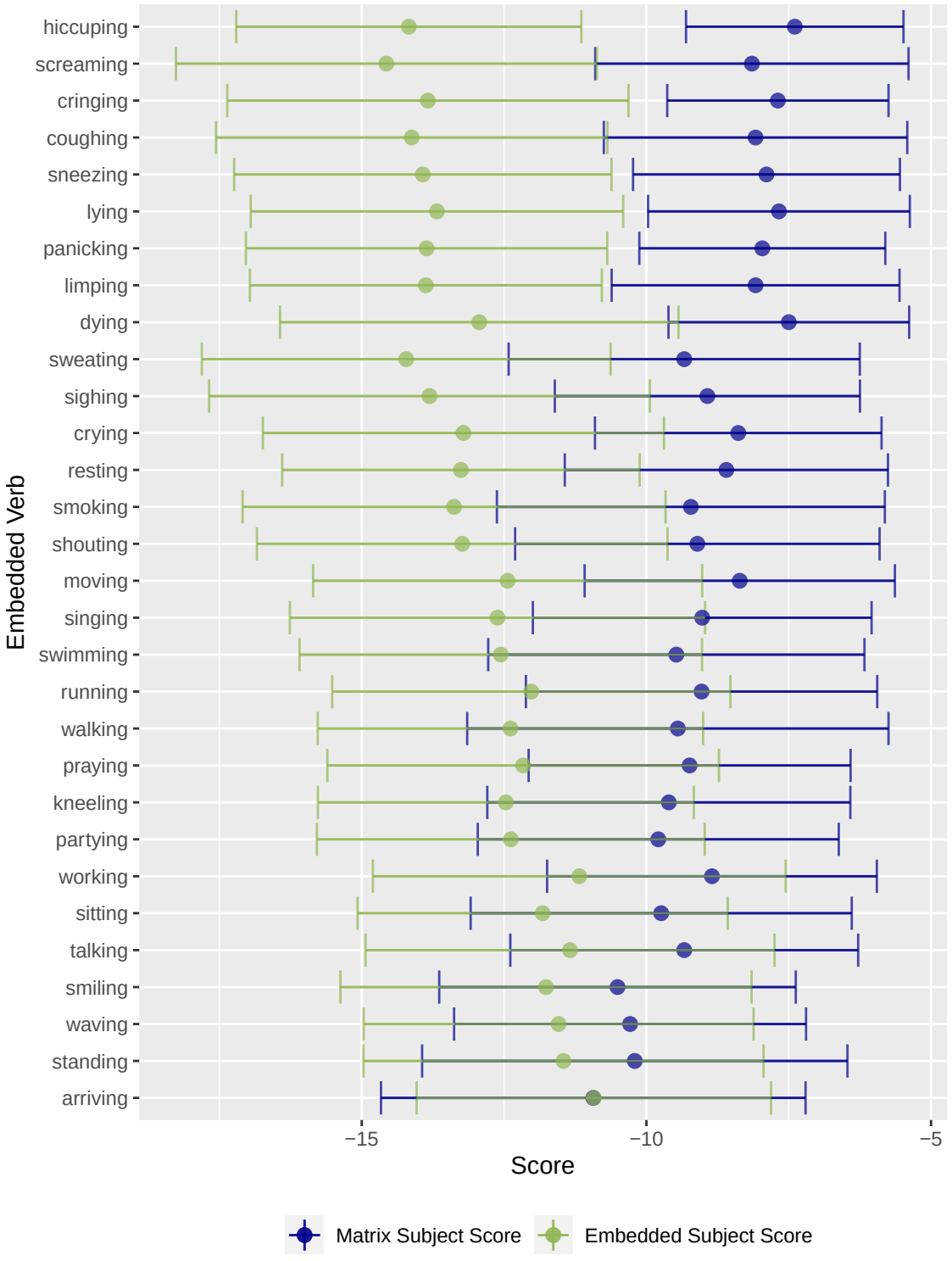


Figure 9: Error bar plot showing mean matrix subject score and embedded subject score for stimuli with each embedded verb. Rows are ordered by matrix subject bias. Error bars show standard deviation.

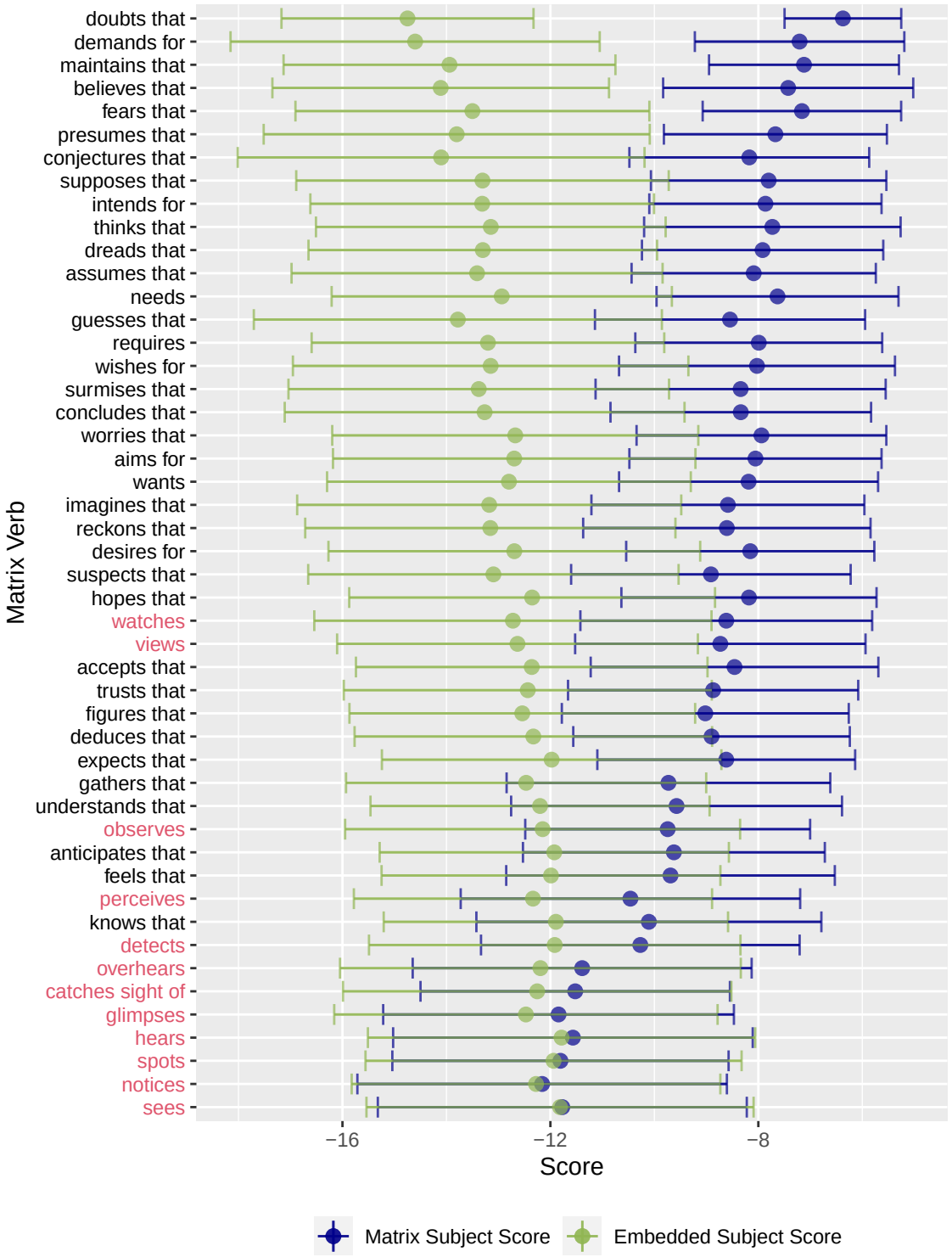


Figure 10: Error bar plot showing mean matrix subject score and embedded subject score for stimuli with each matrix verb. Rows are ordered by matrix subject bias. Error bars show standard deviation. Perceptual matrix verbs are highlighted in red.

Extending Finite-state Models of Reduplication to Tone in Thai

Casey D. Miller
Dept. of Linguistics
University of Utah
u1337847@utah.edu

Aniello De Santo
Dept. of Linguistics
University of Utah
aniello.desanto@utah.edu

Abstract

Languages exhibiting both tonal and reduplication processes pose a challenge for finite-state technologies. In this sense, Markowska et al. (2021) propose a combination of 2-way FSTs and multi-tape FSTs in order to simultaneously deal with total reduplication on the segmental level and independent tonal processes on the autosegmental level. Here, we evaluate this model for reduplication processes in Thai, which shows total reduplication both for tones and segments, and we suggest that the expressivity of 2-way FSTs is needed at both levels.

1 Introduction

Reduplication, the systematic copying/repetition of linguistic content to function with some new grammatical purpose, is a well-attested phenomenon cross-linguistically (Hurch and Mattes, 2005; Rubino, 2005; Raimy, 2012). For instance, Rubino (2005) surveys 368 languages and shows that about 85% exhibit some form of productive reduplication. While the typology of reduplication types is rich, two broader classes of processes have been usually distinguished (Inkelas and Downing, 2015; Urbanczyk, 2007):

- partial reduplication, in which a bounded number of segment are repeated (e.g. the last syllable of a word);
- total reduplication, which repeats unboundedly many segments to form some new morphological constituent.

It has been observed that reduplication presents an interesting challenge to finite-state computational approaches to morpho-phonology (Dolatian and Heinz, 2019b; Rawski et al., 2023). From a computational perspective, by its bounded nature partial reduplication can be modelled with (subsequential) 1-way finite-state transducers (FSTs), although with a significant explosion in the number of required states (Roark and Sproat, 2007). On the other hand, because the number of copied elements has hypothetically no upper bound, total reduplication cannot be modelled with these machines at all — leading some practitioners to adopt memorized lists of words as a way to deal with it in practical applications (Roark and Sproat, 2007; Dolatian and Heinz, 2019a). As total reduplication seems to be one of the few (if not the only) morpho-phonological processes not easily dealt with via 1-way FSTs, it is of particular interest both for practical and theoretical research on finite-state computational models (Dolatian and Heinz, 2019b). In this sense, Dolatian and Heinz (2020) demonstrate how it is possible to use Deterministic 2-way FSTs — essentially, FSTs able to move back and forth on the input tape — to succinctly model both partial and full *segmental* reduplication. Expanding on this intuition, Markowska et al. (2021) observe that a complete finite-state treatment of reduplication cross-linguistically is further complicated by the fact that many languages exhibiting total reduplication are also tonal, and models need to simultaneously capture the somewhat distinct processes affecting the segmental and the autosegmental levels. Importantly, by showing that tones may act independently from their tone-bearing units, classical work in autosegmental phonology has argued for the representational separation of tones from segments (Leben, 1973; Goldsmith, 1976, a.o.). Following work by Dolatian and Rawski (2020), Markowska et al. (2021) argue that modelling the morpho-phonology of languages with both reduplication and tone requires the synthesis of 1-way, 2-way FSTs, and multi-tape FSTs (Filiot and Reynier, 2016; Furia, 2012; Rawski and Dolatian, 2020) — finite state machines with multiple input/output tapes that can be used to mimic autosegmental representations (i.e., splitting the segmental and tonal levels; Wiebe, 1992; Rawski and Dolatian, 2020, a.o.).

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 85-94.
Amherst, Massachusetts, June 15-17, 2023

Importantly, the model in Markowska et al. (2021) is motivated and validated on languages like Shupamem, which exhibit a clear separation between tonal and segmental processes, and that seem to exhibit reduplication only on the segmental level. However, broadening our typological observations is crucial in getting insights into the generalizability of our computational approaches.

Here, we adopt Markowska et al. (2021)'s synthesis approach to reduplication in Thai, building on the observation Thai's total reduplication affects both levels of representation. In other words, Thai exhibits total reduplication both at the segmental and tonal levels, each level then undergoing additional separate transformations (e.g. vowel change in the reduplicant). We then suggest that the approach in Markowska et al. (2021) can be easily extended to languages like Thai by adopting 2-way FSTs for reduplication on both levels, supporting the overall generalizability of the synthetic approach.

2 Reduplication and Tone in Thai

Thai is a member of the Tai-Kadai language family and is the official language of Thailand (Chakshuraksha, 1994). It features five tones (Lee, 2011), which we represent orthographically with diacritics on vowels, following similar literature on the topic: Mid (M; represented by an unmarked V), Low (L; diacritic V̀), High (H; diacritic V́), Rising (R; diacritic V̌), Falling (F; diacritic V̂). Note that for simplicity, we chose to not represent rising and falling tones as a sequence of LH and HL tones, respectively, but this is a choice that does not particularly affect our analysis.[1] Before moving on to a discussion of the variety of reduplication processes available in Thai, we briefly touch on its strict relation between tone preassociation and syllable structure.

2.1 Constraints on Syllable Structure

Thai has a relatively restricted syllable structure: an initial consonant followed by an optional liquid/glide consonant forms the onset, followed by a vocalic nucleus with a tone, and an optional stop/nasal coda (Gandour, 1974; Chakshuraksha, 1994; Hudak, 2007). The general syllable structure, adapted from Cooke (1963), is shown in 1 and 2, the interpretation for which is given in 3, accounting for the phoneme inventory of the language.

1. $C(C_1)\overset{T}{V}(C_2)$

2. $C(C_1)\overset{T}{V\text{ː}}(C_2)$

3. C = any consonant
   $C_1 = \{w, l, r\}$
   $C_2 = \{m, n, ŋ, j, w, p, t, k, ʔ\}$
   V = any vowel
   Vː = any long vowel or the diphthongs /ia/, /ua/, /ɯa/
   T = any tone

In what follows we will ignore the fact that some coda obstruents (C_2) are realized as unreleased {pˀ, tˀ, kˀ}, since this is a transformation not relevant to the process of interest. Note also that vowel length and aspiration are contrastive in Thai, and we use the ː symbol to indicate vowel length.

Thai's tonal phonotactics distinguishes *live* and *dead* syllables. Live syllables are defined as those that end in a sonorant, e.g. [maː] 'to come' or [jàj] 'big'. These are unrestricted and can feature all five tones. Dead syllables are defined as those that end in a stop, e.g. [jàːk] 'to want' or [rót] 'car'. These are restricted: dead syllables with a *short* vowel can feature only low and high tones, while dead syllables with a *long* vowel can feature only low and falling tones. Note that the terms *live* and *dead* are replaced elsewhere in the literature by the terms *unchecked* and *checked*, or *unclosed* and *closed* (Gandour, 1974; Lee, 2011; Cooke, 1963). These constraints on tone showcase the importance of preassociation between segmental and autosegmental levels, and how this might feed into other downstream processes. Thus, attention must be paid when formulating models that posit a strict separation between the two levels of representation (Lee, 2011; Gandour, 1974; Moren and Zsiga, 2001; Rawski and Dolatian, 2020).

2.2 Thai Reduplication

Reduplication in Thai is a productive process that is able to target every grammatical word category (Chakshuraksha, 1994; Sookgasem, 1997). Crucially, total reduplication targets both the segmental and the autosegmental level. We distinguish four types of total reduplication processes, based on their grammatical/semantic function and morpho-phonological changes they induce. This

[1] We follow past work in using an alphabet enriched with diacritics to represent associations between tones and segments, but it is important to keep in mind that enriched alphabets reveal the need for more expressive representations (e.g., graphs) to capture tone beyond orthographic conventions (Yli-Jyrä, 2013; Jardine, 2019).

paper adopts the naming conventions defined in Sookgasem (1997) for the various reduplication patterns: *Simple*, *Complex Type 1*, *Complex Type 2*, and *Complex Type 3*. Complex Type 3 is also called "emphatic reduplication" elsewhere in the literature (Lee, 2011; Haas, 1946; Chakshuraksha, 1994, a.o.). Henceforth, we use the ∼ symbol to separate the base from the reduplicant and represent the reduplication boundary, consistently with Markowska et al. (2021).

2.2.1 Simple Reduplication

Simple Reduplication exhibits no change to the base or reduplicant, neither on the segmental level nor on the tonal level (Sookgasem, 1997; Chakshuraksha, 1994; Haas, 1946). In this type of reduplication the base is copied once and the meaning is changed depending on the word class, as in (i) and (ii).

(i) dɛ̀k→dɛ̀k∼dɛ̀k 'child'→'children'

(ii) nâŋ→nâŋ∼nâŋ
'to sit'→'to sit continuously'

2.2.2 Complex Reduplication Type 1

In Complex Reduplication Type 1 the final vowel of the reduplicant is changed to either /ə/ or /æ/ (iii), both vowels being used interchangeably and usage depends only on speaker preference (Chakshuraksha, 1994; Sookgasem, 1997).

(iii) faŋ→faŋ∼f**æ**ŋ 'to listen'→'to listen'

The autosegmental level is once again fully reduplicated without any changes (in (iii), a mid-tone V is copied as a mid-tone V). This reduplication pattern indicates a level of negativity or disinterest towards something or someone.

2.2.3 Complex Reduplication Type 2

Complex Reduplication Type 2 follows a reduplicant∼base template, with the reduplicant as the first copy, and it is similar in meaning to Complex Reduplication Type 1 (Sookgasem, 1997).

(iv) còt.mă:j→còt.m**ŏ:ŋ**∼còt.m**ă:j**
'a letter'→'a letter'

(v) sì̀t→s**òk**∼s**ì̀t** 'a right'→'a right'

(vi) kà?.t^{h}í?→kà?.t^{h}**ó?**∼kà?.t^{h}**í?**
'coconut milk'→'(something like) coconut milk'

At the segmental level, if the base word ends in /oŋ/, /ok/, or /o?/, then that word cannot undergo this type of reduplication (Sookgasem, 1997). In the reduplicated form, the final syllable of the reduplicant is changed to /oŋ/, /ok/, or /o?/, with the vowel length of the final syllable of the base being maintained. The ending /oŋ/ is used when the final syllable of the base ends in /m/, /n/, /j/, /w/, or in a long vowel — i.e. live syllables (iv). The ending /ok/ is used when the final syllable of the base ends in /p/ or /t/ (v). The ending /o?/ is used when the final syllable of the base is a short vowel followed by a glottal stop (vi). Again, the tonal level is fully reduplicated with no changes.

2.2.4 Complex Reduplication Type 3

Complex Reduplication Type 3 is similar to Simple Reduplication, except that the first copy is made to exhibit a high tone on its final syllable (Sookgasem, 1997; Lee, 2011; Chakshuraksha, 1994; Haas, 1946).

(vii) sŭaj→sú̋aj∼sŭaj
'pretty'→'really pretty'

(viii) nâ:.rák→nâ:.rǎ̋k∼nâ:.rák
'cute'→'really cute'

When the final syllable of the base word already exhibits a high tone, then an *extra* high tone is used (represented with the diacritic V̋). The extra high tone, also called the *emphatic* high tone, is not considered among the basic five tones in Thai because it is not contrasting. Phonetically speaking, the emphatic high tone differs from the basic high tone in that it is higher in pitch and usually lengthened (Lee, 2011). Complex Reduplication Type 3 is, by implication, emphatic or intensifying in meaning.

3 Finite-state Models of Total Reduplication in Tonal Languages

With an understanding of Thai tonal and reduplicative processes in place, in this section we provide a brief, intuitive overview to the classes of finite-state machines combined by Markowska et al. (2021) in their model of total reduplication. We will then explore how this model can be adapted to Thai in the next section.

3.1 Total Reduplication with 2-way FSTs

As mentioned, reduplication in general has been the focus of many studies in the computational linguistics' literature, as it seems to be (one of) the

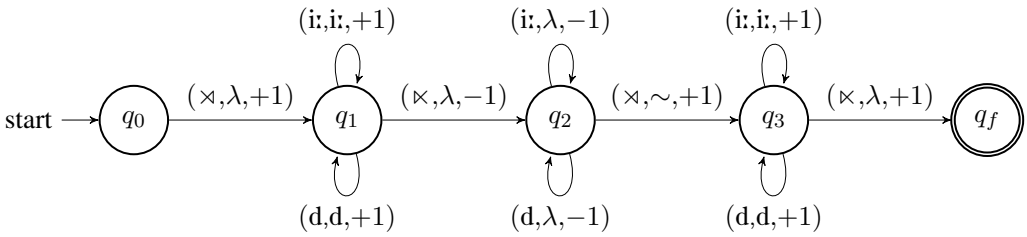


Figure 1: 2-way FST for full reduplication of diː 'good' → diː∼diː 'very good'

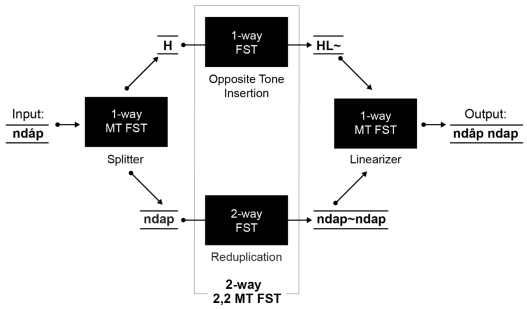


Figure 2: Shupamem reduplication model adapted from (Markowska et al., 2021)

only process(es) in morpho-phonology that cannot be modelled with a 1-way FST (i.e., the output of this process is not a regular language; Roark and Sproat, 2007). In the case of partial reduplication (where only a bounded set of elements needs to be copied) the issue lies in an explosion in the number of states. However, total reduplication affects elements (e.g. full words or phrases) with no a-priory bounds. Dolatian and Heinz (2020) address this problem by adopting 2-way FSTs. Essentially, a 2-way FST increases the expressivity of 1-way FSTs by being able to move back and forth on the input tape, allowing it to read its input more than once (Rabin and Scott, 1959). In designing the machine, state transitions are enriched with a direction parameter ($\{-1,0,+1\}$) that indicates if the FST should move back to the previous symbol, stay on the current symbol, or advance to the next symbol. Dolatian and Heinz (2020) show that this class of transducers not only is able to capture both partial and total reduplication, but it does so in a way that is more transparent with respect to the generalizations argued for in the linguistic literature (see also Dolatian and Heinz, 2019a).

Modelling total reduplication with a 2-way FST involves three steps: (1) reading the input tape left-to-right and outputting the first copy, (2) reading the input tape right-to-left and stopping once the left word boundary ⋊ is read, (3) reading the input tape from left-to-right and outputting the second copy. Figure 1 is an example of a 2-way FST that fully reduplicates the Thai word *diː* 'good' to produce *diː∼diː* 'very good'. In the graphical representation, the input-output pair is grouped with the direction parameter, with each element being separated by a comma. Following Dolatian and Heinz (2019a), we make it so that when reading left-to-right (forward) the input tape is copied on the output tape faithfully. When moving backward (right-to-left), the machine outputs an empty symbol, so that the input string can then be copied again in an additional forward pass. We refer the reader to Dolatian and Heinz (2020) for a full formal treatment of these machines.

3.2 Tone, Reduplication, (2-way) MT FSTs

While the 2-way FST approach of Dolatian and Heinz (2019a) is successful in modeling reduplication at the segmental level, Markowska et al. (2021) point out that many of the world languages exhibiting productive reduplication processes are *tonal*. This presents an additional challenge for finite-state models, as there is the need to handle processes that affect the segmental and autosegmental representations separately. Autosegmental processes have also been argued to exhibit different computational properties than their segmental counterparts (Yli-Jyrä, 2013; Jardine, 2015, 2019, a.o.).

In order to mimic the representational difference between segmental and autosegmental levels within finite-state machines, Dolatian and Rawski (2020) adopt *multi-tape* FSTs (MT FSTs) (see also Fischer, 1965; Wiebe, 1992; Frougny and Sakarovitch, 1993; Furia, 2012; Rawski and Dolatian, 2020). We refer the reader to (Dolatian and Rawski, 2020; Rawski and Dolatian, 2020) for a complete formal treatment of these machines, and here we just cover the basic intuition behind them. Essentially, a MT FST is similar to a 1-way FST with a single tape, but

is able to operate (read from and write to) multiple tapes. This means that such machines can take as input two tapes — a tonal tape and a segmental tape — and operate over them synchronously even when they are subject to different processes.

Using as a motivating starting point Shupamem (a Bantu language), Markowska et al. (2021) observes that a combination of the properties of both 2-way FSTs and MT FSTs is in fact needed to correctly account for the patterns observed in tonal languages with reduplication. Specifically, they synthesize the work in Dolatian and Heinz (2020) and Dolatian and Rawski (2020) to propose deterministic 2-way (n,m) MT FSTs, where n,m refer respectively to the number of input and output tapes. They then present a model of reduplication that makes use of 1-way MT FSTs with a single input tape and two output tapes, in order to split a single string — where tone is orthographically represented with an enriched alphabet using diacritics — into a tonal level and an segmental level. Those are then used as inputs to a 2-way (2,2) MT FSTs composed of a 2-way FST which reduplicates the segmental level, and a 1-way FST dealing with an insertion process on the tonal level. Finally, the two output tapes in the previous step are fed into a $(2,1)$ MT FST which combines them into a reduplicated, enriched output string (Figure 2). Again, we refer the reader to Markowska et al. (2021) for a full discussion of the formal details.

4 Modeling Thai

The synthetic approach surveyed above shows how it is possible to handle both reduplication and autosegmental representations deterministically within a finite-state model. Importantly though, Shupamem (and the other tonal languages analyzed by Markowska et al., 2021) exhibits full reduplication exclusively at the segmental level, while the autosegmental level is affected by other phonological processes targeting tone. Because of this, their 2-way (2,2) MT FST is really 2-way only on one of the two tapes. However, we observed how in Thai the reduplication process on the tonal level mimics the reduplication process on the segmental level. Each of the reduplication types above illustrates full reduplication on both levels, which would by itself be challenging for the single 2-way FST adopted for Shupamem. Additionally, different reduplication types are distinguished by the need of additional dedicated transformations on either the segmental or autosegmental level. Specifically, Complex Reduplication Type 2 showcases transformations that target segmental information, while Type 3 illustrate changes targeting tone specifically.

C	any consonant
V	any vowel
T	any tone
T′	{M, L, R, F}
K	{p, t, k, ʔ}
S	{m, n, ŋ, j, w}
C′	C - S
E	extra high tone
λ	empty string

Table 1: List of shorthand symbols used in the FSTs.

Because of these facts, Thai serves as a good test case to explore the flexibility of the synthetic approach. In particular, by formalizing the reduplication types discussed above, in what follows we illustrate how Thai clearly shows the need for 2-way FSTs on both segmental and autosegmental tapes.

We assume a model like the one in Figure 2, which utilizes MT FSTs as *splitters* and *linearizers* to move from and to orthographic representations with an enriched alphabet. These MT FSTs are unchanged with respect to the ones presented by Markowska et al. (2021), and thus we refrain from including examples of them in this paper. We focus instead on the application of the 2-way $(2,2)$ MT FST (boxed section in Figure 2) to the variety of reduplication processes in Thai.

Henceforth, we define the alphabet our machines operate on using the following shorthand: C refers to any consonant, V refers to any short vowel, V: refers to any long vowel or diphthong, and a period (.) to syllable boundaries. Additionally, we use K for the set {p, t, k, ʔ}, and S for the set {m, n, ŋ, j, w}. A summary of these abbreviations (and all those used in the FSTs that follow) is shown in Table 1.

4.1 Syllable-Tone Association

If we follow Markowska et al. (2021)'s in adopting an initial alphabet with diacritics, it seems useful to incorporate an additional step before the splitter in order to guarantee the correct preassociations of tones and segments. Recall that tone restrictions are placed only on dead syllables: short dead syllables only feature low and high tones, and long dead syllables only feature low and falling tones. As these constraints are all local over the enriched alphabet, we could easily handle them

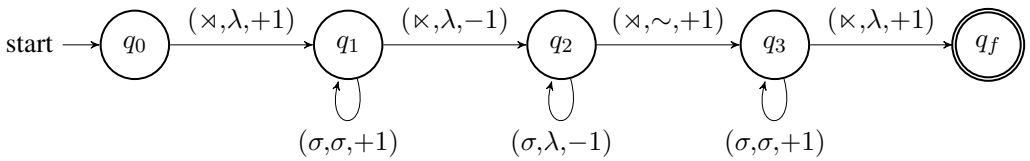


Figure 3: 2-way FST for Simple Reduplication (either segmental or tonal level).

with a 1-way FST. Dealing with tonal constraints with 1-way FSTs over enriched representations in not novel of course (see for example Yli-Jyrä, 2013, a.o.), and we could alternatively handle preassociation with MT FSTs scanning the two levels synchronously (Rawski and Dolatian, 2020). What this draws attention to though, is the need to consider tone-segment preassociation even within models which require separate levels at some point.

4.2 Simple Reduplication Model

We can now start looking at Thai's reduplication processes. Recall that in the case of simple reduplication, both the segmental and autosegmental levels undergo total reduplication, with both copies being rendered faithfully with respect to the input:

sàʔ.ʔàːt→sàʔ.ʔàːt∼sàʔ.ʔàːt
'clean'→'very clean'

Although the synthetic model for Shupamem assumes a 1-way FST for tone, the most general, formal definition of 2-way (2, 2) MT FST in Markowska et al. (2021) seems to allow for 2-way FSTs on both tapes. This is exactly the approach that we take. Figure 3 is an example of 2-way FST that models simple reduplication in Thai. This is essentially identical to the FST shown in Figure 1. The symbol σ represents any symbol in an alphabet, that is $\sigma \in \Sigma$, so that (instances of) this FST can work for both the segmental level and the tonal level. A (2,2) MT FST of simple reduplication would then apply an instantiation of the FST in Figure 3 on both tapes.

4.3 Complex Reduplication Type 1

Consider now Complex Reduplication of Type 1:

faŋ→faŋ∼fæŋ 'to listen'→'to listen'

Recall that a vowel without a diacritic is not toneless, but bears a Mid tone. This reduplication type shows full reduplication of both tones and segments, but at the segmental level the final vowel of the reduplicant is changed to either /ə/ or /æ/ (we will use /æ/ for simplicity, since this assignment is speaker-specific).

A 2-way FST that reduplicates the segmental level is shown in Figure 4, a derivation for which is shown in Table 2. The first time the word is copied, it is copied faithfully. The second time it is copied, we want the final vowel of the word to change. For this reason, we output the syllable and loop back to q_3 until a word boundary symbol is read. Once the word boundary symbol is read, the final syllable is outputted accordingly, including the vowel change. For total reduplication on the tonal level, the FST in Figure 3 suffices since there is no tone change.

State	Input-Tape		Output-Tape
q_0	⋊*sa*ʔ.ʔa:*t*⋉	+1	λ
q_1	⋊*s*a*ʔ*.ʔa:*t*⋉	+1	s
q_1	⋊*sa*ʔ.ʔa:*t*⋉	+1	sa
q_1	⋊*sa*ʔ.ʔa:*t*⋉	+1	saʔ
q_1	⋊*sa*ʔ.ʔa:*t*⋉	+1	saʔ.
q_1	⋊*sa*ʔ.*ʔ*a:*t*⋉	+1	saʔ.ʔ
q_1	⋊*sa*ʔ.ʔ*a*:*t*⋉	+1	saʔ.ʔa:
q_1	⋊*sa*ʔ.ʔa:*t*⋉	+1	saʔ.ʔa:t
q_1	⋊*sa*ʔ.ʔa:*t*⋉	−1	saʔ.ʔa:t
q_2	⋊*sa*ʔ.ʔa:*t*⋉	−1	saʔ.ʔa:t
q_2	⋊*sa*ʔ.ʔ*a*:*t*⋉	−1	saʔ.ʔa:t
q_2	⋊*sa*ʔ.*ʔ*a:*t*⋉	−1	saʔ.ʔa:t
q_2	⋊*sa*ʔ.ʔa:*t*⋉	−1	saʔ.ʔa:t
q_2	⋊*sa*ʔ.ʔa:*t*⋉	−1	saʔ.ʔa:t
q_2	⋊*s*a*ʔ*.ʔa:*t*⋉	−1	saʔ.ʔa:t
q_2	⋊*sa*ʔ.ʔa:*t*⋉	−1	saʔ.ʔa:t
q_2	⋊*sa*ʔ.ʔa:*t*⋉	+1	saʔ.ʔa:t∼
q_3	⋊*s*a*ʔ*.ʔa:*t*⋉	+1	saʔ.ʔa:t∼s
q_4	⋊*s*a*ʔ*.ʔa:*t*⋉	+1	saʔ.ʔa:t∼s
q_6	⋊*sa*ʔ.ʔa:*t*⋉	+1	saʔ.ʔa:t∼s
q_8	⋊*sa*ʔ.ʔa:*t*⋉	+1	saʔ.ʔa:t∼saʔ.
q_3	⋊*sa*ʔ.*ʔ*a:*t*⋉	+1	saʔ.ʔa:t∼saʔ.ʔ
q_4	⋊*sa*ʔ.ʔ*a*:*t*⋉	+1	saʔ.ʔa:t∼saʔ.ʔ
q_5	⋊*sa*ʔ.ʔa:*t*⋉	+1	saʔ.ʔa:t∼saʔ.ʔ
q_7	⋊*sa*ʔ.ʔa:*t*⋉	+1	saʔ.ʔa:t∼saʔ.ʔæ:t

Table 2: Complex Type 1 derivation for the segmental level (Figure 4) of *sàʔ.ʔàːt* 'clean' → *sàʔ.ʔàːt∼sàʔ.ʔæ̀ːt* 'too clean'.

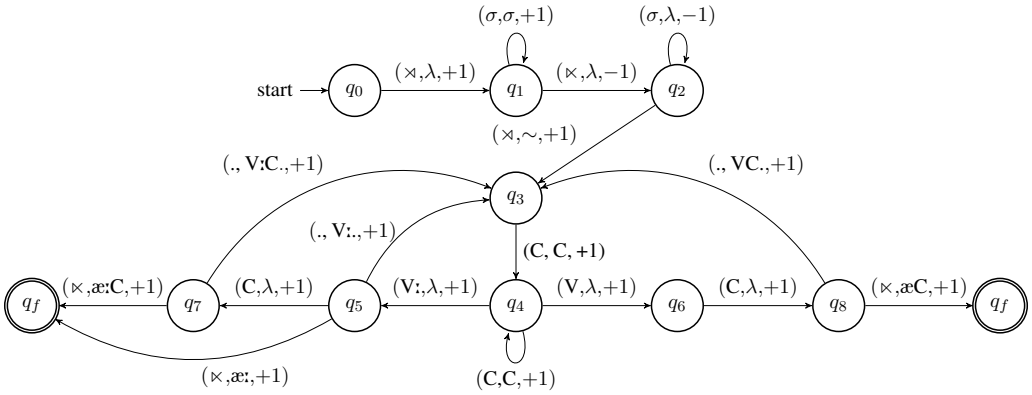


Figure 4: 2-way FST for Complex Reduplication Type 1 at the segmental level.

4.3.1 Complex Reduplication Type 2

Complex Reduplication of type 2 involves a reduplicant-base pattern, with a change to the final sylllable of the reduplicant (the first copy):

> càt.măːj→càt.m**ŏːŋ**∼càt.m**ăːj**
> 'a letter'→'a letter'

An FST that handles reduplication for the segmental level for Complex Reduplication Type 2 is shown in Figure 5. For the sake of readability, only one of the three endings (/oŋ/) is considered here. We use S as a shorthand for the set {m, n, j, w}. The shorthand C represents the set of all consonants in Thai, as previously used in this paper. The shorthand C′ represents the set of all consonants in Thai excluding the set S, such that the operation C−S = C′ holds true.

For this process, the first time a word is copied we want the rhyme of the final syllable to change. Thus, we loop back to q_1 until a word boundary symbol is read. The FST only allows words to end in consonants in the set S = {m, n, j, w}. Once the first copy is outputted with the rhyme change, then the second copy is faithfully read and outputted.

We mentioned that Complex Reduplication Type 2 is not possible for words that end in /oŋ/, /ok/, or /oʔ/ (Sookgasem, 1997). We could of course include this restriction in the FST in Figure 5, for example by handling the /o/ and /oː/ vowels separately from all other vowels, and excluding a transition where the ⋉ symbol is read after a syllable containing /o/ or /oː/. Alternatively, another FST could be added to the pipeline to filter what kind of inputs are appropriate for each reduplication type. Once again, we can use the FST in Figure 3 for the tonal level reduplication here since it involves total reduplication with no tone change.

4.3.2 Complex Reduplication Type 3

In Complex Reduplication of type 3, the segmental level is reduplicated faithfully (which can be accomplished with the FST in Figure 3). At the autosegmental level, the final syllable of the first copy is made to bear a high tone, while the original tone appears faithfully in the second copy:

> nâː.rák→nâː.rǎk∼nâː.rák
> 'cute'→'really cute'

This process is modelled by the 2-way FST in Figure 6. We use T as a stand in for any tone ({M, L, H, R, F}) except for the extra high tone with we represent as E, and T′ to stand in for non-high tones ({M, L, R, F}). For the first copy, as the only tone that needs to be changed is associated to its last syllable, after reading a tone from the input tape the FST "waits" to check whether the immediate next element is a boundary symbol (⋉) before outputting it. If the tone was a non-high tone and the next element is ⋉, a high tone is outputted. If the tone was a high tone and the next element is ⋉, then an extra-high tone is outputted. If not at the end of the string, tones are outputted faithfully. The second copy is fully faithful.

5 Conclusion

This paper builds on previous work in adopting a deterministic finite-state approach to model the interaction of total reduplication and tonal processes in Thai. Markowska et al. (2021) synthesized an approach to autosegmental processes via MT FSTs (Dolatian and Rawski, 2020; Rawski and Dolatian, 2020) and 2-way FSTs to deal with total reduplication (Dolatian and Heinz, 2019a, 2020) in order to account for what observed in Shupamem. They show how this combination allows them to deal with

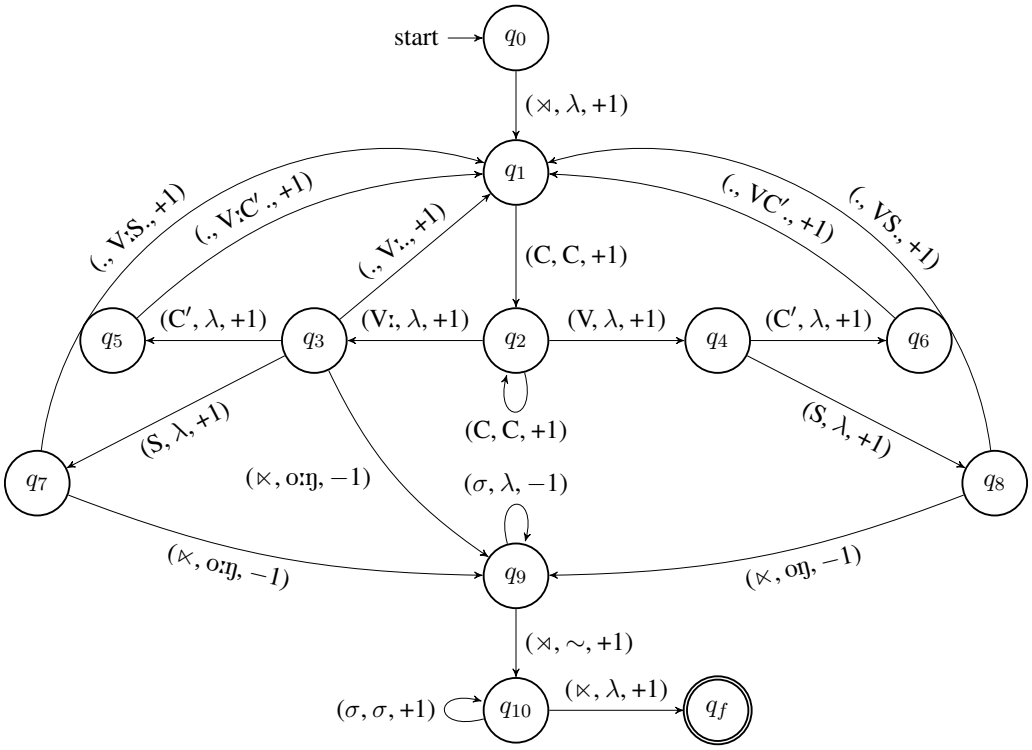


Figure 5: 2-way FST for the segmental level of Complex Reduplication Type 2.

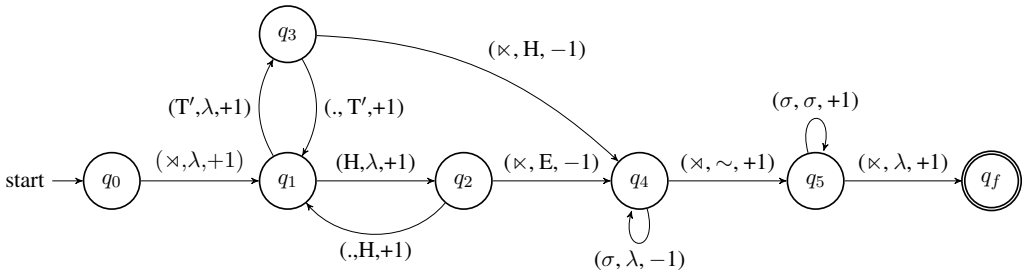


Figure 6: 2-way FST for the tonal level of Complex Reduplication Type 3.

the double challenge of handling unbounded copies (as required by total reduplication), and separate segmental and autosegmental processes while remaining faithful to linguistic analyses of these patterns.

Crucially, Shupamem exhibits total reduplication exclusively on the segmental level, thus allowing the model to fully treat tone and segments separately. Here, we used Thai as an example of a language where tones also undergo reduplication. We suggested then to take full advantage of the expressivity of the 2-way (2,2) MF FST model, by making sure that both the segmental and the autosegmental tapes are used as inputs to 2-way FSTs. In doing this, we showed how carefully exploring the typological diversity of tonal languages with reduplication will enrich our understanding of the expressivity required by finite-state models.

Looking back at our analyses of Thai, it is reasonable to wonder whether we could have handled the reduplication pattern as a whole with a single 2-way FST, without need for the MT FST split. While this is doable adopting an enriched alphabet, the MT FST approach allows us to remain as close as possible to linguistic analyses when modeling the independent changes the segmental and autosegmental levels go through in the Complex Reduplication types. However, the concatenation of 2-way and multi-tape FSTs potentially pushes the expressivity of these machines quite high (Fischer, 1965; Furia, 2012), stressing how crucial it is going to be for an insightful computational theory of morpho-phonology to conduct an extensive formal

evaluation of the expressive power of alternative combinations/restrictions of these devices.

In sum, these results add support to the deterministic finite-state approach to total reduplication advanced in previous literature, while highlighting the fundamental role of broader typological evaluation.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable feedback. This work was supported by funding from the Undergraduate Research Opportunities Program at the University of Utah awarded to Casey Miller.

References

Nuttathida Chakshuraksha. 1994. Prosodic structure and reduplication in thai. *Working Papers of the Linguistics Circle*, 12:27–38.

Joseph R. Cooke. 1963. The vowels and tones of standard thai: Acoustical measurements and experiments. arthur s. abramson. *American Anthropologist*, 65:1406–1407.

Hossep Dolatian and Jeffrey Heinz. 2019a. Learning reduplication with 2-way finite-state transducers. In *International Conference on Grammatical Inference*, pages 67–80. PMLR.

Hossep Dolatian and Jeffrey Heinz. 2019b. Redtyp: A database of reduplication with computational models. *Proceedings of the Society for Computation in Linguistics*, 2(1):8–18.

Hossep Dolatian and Jeffrey Heinz. 2020. Computing and classifying reduplication with 2-way finite-state transducers. *Journal of Language Modelling*, 8(1):179–250.

Hossep Dolatian and Jonathan Rawski. 2020. Multi-input strictly local functions for templatic morphology. *Proceedings of the Society for Computation in Linguistics*, 3(1):282–296.

Emmanuel Filiot and Pierre-Alain Reynier. 2016. Transducers, logic and algebra for functions of finite words. *ACM SIGLOG News*, 3(3):4–19.

Patrick C Fischer. 1965. Multi-tape and infinite-state automata—a survey. *Communications of the ACM*, 8(12):799–805.

Christiane Frougny and Jacques Sakarovitch. 1993. Synchronized rational relations of finite and infinite words. *Theoretical Computer Science*, 108(1):45–82.

Carlo A Furia. 2012. A survey of multi-tape automata. *arXiv preprint arXiv:1205.0178*.

Jack Gandour. 1974. On the representation of tone in siamese. *UCLA Working Papers in Phonetics*, 27:118–146.

John Anton Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

Mary R Haas. 1946. Techniques of intensifying in thai. *Word*, 2(2):127–130.

Thomas John Hudak. 2007. *William J. Gedney's comparative Tai source book*. University of Hawaii Press.

Bernhard Hurch and Veronika Mattes. 2005. *Studies on reduplication*. Mouton de Gruyter Berlin.

Sharon Inkelas and Laura J Downing. 2015. What is reduplication? typology and analysis part 1/2: The typology of reduplication. *Language and linguistics compass*, 9(12):502–515.

Adam Jardine. 2015. Computationally, tone is different. *Phonology*.

Adam Jardine. 2019. The expressivity of autosegmental grammars. *Journal of Logic, Language and Information*, 28:9–54.

William Ronald Leben. 1973. *Suprasegmental phonology.* Ph.D. thesis, Massachusetts Institute of Technology.

Leslie Lee. 2011. Fixed autosegmentism in thai emphatic reduplication. *Journal of the Southeast Asain Linguistics Society*, 4:41–63.

Magdalena Markowska, Jeffrey Heinz, and Owen Rambow. 2021. Finite-state model of shupamem reduplication. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 212–221.

Bruce Moren and Elizabeth Zsiga. 2001. lexical tone and markedness in standard thai. In *Annual Meeting of the Berkeley Linguistics Society*, volume 27, pages 181–191.

Michael O Rabin and Dana Scott. 1959. Finite automata and their decision problems. *IBM journal of research and development*, 3(2):114–125.

Eric Raimy. 2012. *The phonology and morphology of reduplication*, volume 52. Walter de Gruyter.

Jonathan Rawski and Hossep Dolatian. 2020. Multi-input strict local functions for tonal phonology. *Proceedings of the Society for Computation in Linguistics*, 3(1):245–260.

Jonathan Rawski, Hossep Dolatian, Jeffrey Heinz, and Eric Raimy. 2023. Regular and polyregular theories of reduplication. *Glossa: a journal of general linguistics*, 8(1).

Brian Roark and Richard Sproat. 2007. *Computational approaches to morphology and syntax*, volume 4. OUP Oxford.

Carl Rubino. 2005. Reduplication: Form, function and distribution. *Studies on reduplication*, 28:11–29.

Prapa Sookgasem. 1997. A complicating distortion of syntactic categories: The case of reduplication in thai. *Southeast Asian linguistics studies in honor of Vichin Panupong*, pages 253–272.

Suzanne Urbanczyk. 2007. Themes in phonology. *The Cambridge Handbook of Phonology, edited by Paul de Lacy*, pages 473–493.

Bruce Wiebe. 1992. Modelling autosegmental phonology with multi-tape finite state transducers.

Anssi Mikael Yli-Jyrä. 2013. On finite-state tonology with autosegmental representations. In *Proceedings of the 11th international conference on finite state methods and natural language processing*. The Association for Computational Linguistics.

A Appendix

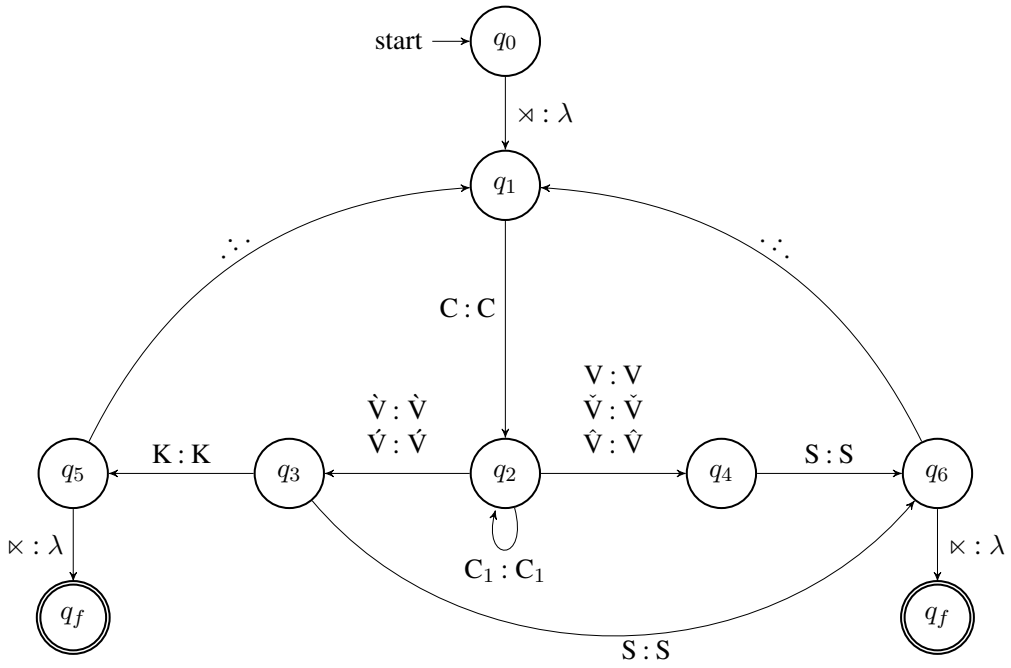


Figure 7: 1-way FST to model the phonotactics of short dead syllables in Thai. $C_1 = \{w, l, r\}$.

Extracting binary features from speech production errors and perceptual confusions using Redundancy-Corrected Transmission

Zhanao Fu
University of Toronto
zhanao.fu@mail.utoronto.ca

Ewan Dunbar
University of Toronto
ewan.dunbar@utoronto.ca

Abstract

We develop a mutual information-based feature extraction method and apply it to English speech production and perception error data. The extracted features show different phoneme groupings than conventional phonological features, especially in the place features. We evaluate how well the extracted features can define natural classes to account for English phonological patterns. The features extracted from production errors had performance close to conventional phonological features, while the features extracted from perception errors performed worse. The study shows that featural information can be extracted from underused sources of data such as confusion matrices of production and perception errors, and the results suggest that phonological patterning is more closely related to natural production errors than to perception errors in noisy speech.

1 Introduction

Phonological features have usually been assumed to be phonetically grounded in addition to explaining phonological behaviour. Yet the sources of phonetic data that have been used to infer the nature of phonological features are largely limited to physical acoustic and articulatory measures. Furthermore, the analytical methods available to infer features that are consistent with phonetic data are limited. This study proposes a new method for automatically inferring binary features from similarity matrices, which lends itself to directly studying data relevant to human phonetic processing: here we study perception and production errors.

Previous work has attempted to infer phonetically-grounded features using clustering (Lin, 2005; Lin and Mielke, 2006; Mielke, 2008, 2012; Shain and Elsner, 2019). For example, Mielke (2008) modelled consonant similarity using hierarchical clustering applied to perceptual confusion data, which combines consonants together into nested clusters.

However, clustering does not directly output features in the usual sense of independent, cross-cutting properties of phonemes. Non-hierarchical clustering applied to phonemes yields a flat set of classes, the equivalent of a single binary or n-ary feature. Hierarchical clustering yields classes that can contain other class divisions (for example, a cluster of vowels can be subdivided into a cluster of high and a cluster of low vowels, and so on). However, in typical approaches to hierarchical clustering, decisions as to how to make sub-clusters are taken independently in each cluster. Features are thus not allowed to have scope over more than one sub-cluster. Not only does this contrast sharply with usual approaches to phonological features which naturally give rise to parallel relations across clusters—the "proportional oppositions" of Trubetzkoy (1969)—it means that any data about similarity between phonemes across clusters is necessarily ignored by such algorithms.

To address these issues, we develop a method inspired by Miller and Nicely's (1955) analysis of confusion matrices, based on an information-theoretic measure of *feature transmission*. We first introduce the algorithm and demonstrate it using an artificial example. Next, we report an experiment where the feature extraction algorithm is applied to phoneme perception and production errors, and the extracted features sets are evaluated based on their utility and efficiency in describing phonological classes. Finally, we discuss the insights yielded for the study of phonological features.

Although the paper infers phonological features from data, our goal is not to argue that phonological features are emergent. This paper analyzes confusion data, and determines what set of features would be most compatible with the data (under certain assumptions). While this could be consistent with a hypothesis that learners infer features based on their own confusions, we tend toward the opposite view: features are primary, and feature

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 95-107.
Amherst, Massachusetts, June 15-17, 2023

similarity is a *cause* of confusions. In any case, our analysis is correlational, and as such it is neutral to what is the cause and what is the result. The question is merely what features best explain the data at hand.

Of course, if we do assume that feature representations are one cause of errors (rather than assuming that features are emergent from error patterns), we must accept that feature similarity is only one cause among others—for example, noise in the audio signal, the nature of that noise, physiological constraints on production, and phonological neighbourhoods (Vitevitch, 2002), among other things. For our purposes, we need to assume that the effect of distinctive features on error patterns is strong enough to be detected in spite of these other factors.

2 Extracting feature with Redundancy-Corrected Transmission

2.1 Background

Miller and Nicely (1955) analyzed confusion matrices from an identification task in which participants heard a CV syllable in noise (a consonant followed by /ɑ/) and had to provide a phonemic label for the onset consonant. They developed an information-theoretic measure of feature transmission in a confusion matrix, using it as part of an argument that listeners use distinctive features in speech perception.

Miller and Nicely assumed that speech processing works by transmitting information over a fixed number of channels (features). They used five features to analyze English consonants (voicing, nasality, affrication, duration, and place). By analyzing the confusions between consonants with opposing values for each feature separately (e.g., between voiced and voiceless sounds), they measured the amount of information faithfully transmitted for each feature under various amounts of additive noise. They argue that the result of this analysis suggests that each of these five features is perceived by listeners independently of the others, since the sum of the information transmitted for these five features is close to the the amount of transmitted information measured if phonemes are not organized into features—little information is lost by analyzing phonemes into independent features.

For our purposes, it is not this argument that matters but their transmission measure itself, which can be seen as a measure of how "consistent" a hypothetical feature is with a given confusion matrix. In particular, a hypothetical feature which is consistently extremely poorly transmitted is clearly not implicated in perception. In what follows, we develop this intuition further and show its limitations, and motivate the use of a further term penalizing feature **redundancy**. We show that, despite its limitations, the idea of discovering a set of features with high transmission and minimal redundancy leads to satisfactory results in an artificial example.

2.2 Developing the algorithm

We use a hypothetical phoneme mapping process within a 4-phoneme inventory [ABCD] to illustrate these ideas. We assume these phonemes are transmitted via some noisy process (for example, perception or production) whose goal is accurate transmission—in other words, to faithfully map an input phoneme to itself. Table 1 summarizes a possible outcome from repetitions of this transmission process with different input phonemes.

		Input			
		A	B	C	D
Output	A	10	8	2	0
	B	8	10	0	2
	C	2	0	10	8
	D	0	2	8	10

Table 1: A confusion matrix of the hypothetical mappings in a four-phoneme system with two features.

Furthermore, we assume that, in this hypothetical process, the phonemes are transmitted by transmitting the values of two underlying features *f1* ([AB | CD]) and *f2* ([AC | BD]). As features are often transmitted with different degrees of degradation (Miller and Nicely, 1955), we make it so that *f1* is maintained better than *f2*, resulting in more confusions between phoneme pairs that are differentiated by *f2* (such as A and B) than between phoneme pairs differentiated by *f1* (such as A and C). Our goal of feature extraction is to infer the true underlying features (*f1* and *f2*) based only on the confusion matrix. To achieve this, we consider all potential features, i.e., all binary groupings (While nothing prevents the algorithm we develop here from being used with n-ary features, we restrict the current paper to binary features.). We examine how well each potential feature is transmitted by collapsing the confusion matrix according to that feature. We show this in Table 2 for the feature that splits the inventory into [AB | CD] (which happens

to be one of the true features used in transmission).

		Input	
		+ AB	− CD
Output	+ AB	36	4
	− CD	4	36

Table 2: Collapsed confusion matrices for the example in Table 1 according to the feature that splits the inventory into [AB | CD].

Higher counts on the diagonal represent more faithful transmissions in the collapsed confusion matrix. Thus, even at first glance, the feature in Table 2 is a good candidate for a feature which is transmitted faithfully. To quantitatively evaluate how well a feature is preserved in the output, we calculate the *transmission* of a signal from the input (I) to the output (O) with Equation 1 as defined in Miller and Nicely (1955).

$$T(I;O) = \sum_{o \in O} \sum_{i \in I} p(i,o) \log \frac{p(i,o)}{p(i)p(o)} \quad (1)$$

When the confusion matrix is collapsed based on a potential feature f, $T(X_f;Y_f)$ evaluates the how much information about the feature is transmitted.

The transmission alone can capture how much information is transmitted, but it is not sufficient to evaluate how well a feature is transmitted. This is because the transmission value is influenced not only by how well information from the input is preserved in the output, but also by both how much information was contained in the input in the first place. In order to eliminate the influence of the information in the input, we instead evaluate the proportion of the input information successfully transferred to the output. First, we quantify the amount of information in the input by calculating the *entropy* of input re-coded with the feature, as defined in Equation 2:

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \quad (2)$$

in order to calculate the *relative transmission* $T_{rel}(X_f;Y_f)$ of the input information with respect to the feature f:

$$T_{rel}(X_f;Y_f) = \frac{T(X_f;Y_f)}{H(X_f)} \quad (3)$$

in which $H(X_f)$ is the amount of information in the input and $T(X_f;Y_f)$ is the amount of information shared by the input and the output.

With the relative transmission criterion, we can evaluate all possible candidate features to characterize the inventory [ABCD], as seen in Table 3. The relative transmission of the true underlying feature f_1 (feature I in the table), is higher than that of any other hypothetical feature, as expected, given that our constructed transmission process was one in which this feature was well-transmitted.

However, to extract a set of relevant features, simply seeking a set of features in which each feature individually has a high relative transmission would usually not result in an ideal feature set. This is because a highly informative feature can often undergo a minor adjustment to create a slightly different, spurious, feature that also has high transmission. Consider Table 3 again: hypothesized feature II corresponds to the second true underlying feature that was used to generate the example, *f2*. While its relative transmission of 0.029 is higher than that of the (incorrect) feature III, it is still lower than that of features IV and V. These features have a high relative transmission because they largely overlap with the well-transmitted feature *f1*, grouping together either [CD] (feature IV) or [AB] (feature V). In order to avoid extracting features partially containing the information included in already selected feature, we consider the redundancy of the new feature with respect to each old feature by calculating the *mutual information* $I(X;Y)$. The mutual information captures the degree of association between the states of two variables. As such, it can be used to evaluate the similarity between two features. The mutual information $I(X;Y)$ for two discrete random variables X and Y is defined as:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (4)$$

When evaluating the similarity between features, X and Y are the counts of the input variable re-coded with the two features, respectively.

To keep the mutual information between features on the same scale as the relative transmission of features, we also define a *relative mutual information* $I_{rel}(X_{f_a};X_{f_a})$ between the features f_a and f_b to quantify a new feature's redundancy with respect to an existing feature f_a.

$$I_{rel}(X_{f_a};X_{f_b}) = \frac{I(X_{f_a};X_{f_b})}{H(X_{f_a})} \quad (5)$$

Features		I		II		III		IV		V	
Value		+	−	+	−	+	−	+	−	+	−
		AB	CD	AC	BD	AD	BC	A	BCD	C	ABD
+		36	4	24	16	20	20	10	10	10	10
−		4	36	16	24	20	20	10	50	10	50
$T_{rel}(X_f; Y_f)$		0.531		0.029		0		0.091		0.091	
$J(f; S)$	Step 1	**0.531**		0.029		0		0.091		0.091	
	Step 2	\		**0.029**		0		-0.22		-0.22	

Table 3: Evaluating features in the four-phoneme system from Table 1. The table includes collapsed confusion matrices according to different features, the corresponding $T_{rel}(X_f; Y_f)$, and the RCT criterion $J(f; S)$ at two steps of feature extraction. The $J(f; S)$ values of the selected feature at each step are marked in bold. After two steps, the selected features are efficient to differentiate all phonemes and the algorithm ends.

Together this leads us to propose the Redundancy-Corrected Transmission (RCT) criterion $J(f, S)$:

$$J(f; S) = T_{rel}(X_f; Y_f) - \frac{1}{|S|} \sum_{f_i \in S} I_{rel}(X_f; X_{f_i}) \quad (6)$$

In the RCT criterion we use the average of the relative mutual information between the candidate feature (f) and each of the features that are already selected ($f_i \in S$) to minimize redundancy. In addition to this, we also filter the non-contrasting features from candidate feature set before each step of feature selection. Non-contrasting features are defined as the candidate features that do not create new contrast between phonemes given a set of selected features. For example, in a hypothetical consonant inventory [p t f s m n v z], assuming that two features [p t f s | m n v z] ([voice]) and [p t m n | f s v z] ([continuant]) have been selected, then the feature [p t v z | m n f s] would be a non-contrasting feature since it does not create any divisions in the smallest classes (i.e., [p t], [f s], [m n], [v z]) created by the two previous features. This filtering process ensures that the algorithm finds a compact set of features to encode all phonemes.

The extraction process above is summarized in Algorithm 1.

Algorithm 1: Binary feature extraction algorithm with RCT.

Data: A confusion matrix for n items
Result: A set of binary features
$F \leftarrow \varnothing$
for $i = 1$ **to** 2^{n-1} **do**
  $F = F \cup i$ (as a binary string)
end
$S \leftarrow \varnothing$
while *Not all phonemes have distinct featural representations* **do**
  $f_{select} =_{f \in F} J(f, S)$
  $S = S \cup \{f_{select}\}$
  $F = F - \{f_{select}\}$
  for $f \in F$ **do**
    if $|unique(X_{S \cup \{f\}})| = |unique(X_S)|$ **then**
      $F_{redundant} = F \cup \{f\}$)
    end
  end
  $F = F - F_{redundant}$
end

2.3 Preprocessing

Finally, we will discuss the preprocessing steps that are important in the preparation of confusion data for feature extraction. In real data, especially in the errors collected from natural speech, three issues are often present.

First, some input phonemes may present very few errors. The sparsity of the data for a given phoneme means that it may be difficult to distinguish between hypothesized features on the basis of this phoneme. We address this issue by applying add-one smoothing to the data. In add-one smoothing, we take each column in the confusion matrix that corresponds to the counts (number of errors) for the input phoneme, then add one to all the values in the column. Second, in some kinds of data, the number of examples of each phoneme in the input may not be balanced. This is notably the case in speech error data, which is observational. To avoid high-frequency phonemes having an undue influence, we balance the data by con-

verting the matrices of the error counts into the error probability for each phoneme. Summing up these first two steps, we estimate the probability of mapping input phoneme i to output phoneme j as $p_{ij} = (n_{ij} + 1)/((\sum_i n_{ij}) + n_{jj})$.

The third potential issue arises in the speech error data: while the data lists the errors, it does not record counts of the number of correctly articulated instances. Missing faithful transmissions could potentially lead to errors in feature extraction.

		Input		
		x	*y*	*z*
Output	*x*	○	✓	
	y	✓	○	✓
	z		✓	○

Table 4: Confusion matrix for a hypothetical phoneme inventory. Check marks represent confusions phonemes, circles represent faithful mappings. Without the faithful transmissions, *x* and *z* cannot be differentiated.

Consider the example in Table 4, a hypothetical phoneme inventory with three phonemes *x*, *y*, *z*, and two underlying features, one separating *x* and *y* against *z*, the other separating *y* and *z* against *x*. Without the faithful mappings, both *x* and *z* would only have data from confusions with *y*, making it impossible to differentiate *x* and *z*. As a result, the incorrect feature [x z | y] has the highest transmission and would be selected as the first feature. In order to prevent similar issues in the data where faithful mappings are missing, the diagonal of the confusion matrix needs to be filled in before the feature extraction.

In our experiment, we fill the diagonal cells in the confusion matrices with the sum of the error counts in the corresponding column, which results in a 50% error rate for each input phoneme. The 50% error rate provides information of phoneme identity to address the issue described above, while also maintains the contrasts between phonemes.

Table 5 shows the preprocessed data after each step, from the artificial example in Table 1.

3 Experiment

We apply Algorithm 1 to a perceptual confusion matrix from Miller and Nicely (1955), as well as to a collection of speech error data from Fromkin (1971). We evaluate how well the resulting features can be used to define natural classes in English.

	A	B	C	D
A	10	8	2	0
B	8	10	0	2
C	2	0	10	8
D	0	2	8	10

(a) original data

	A	B	C	D
A	11	9	3	1
B	9	11	1	3
C	3	1	11	9
D	1	3	9	11

(b) add-one smoothing

	A	B	C	D
A	11	0.7	0.2	0.1
B	0.7	11	0.1	0.2
C	0.2	0.1	11	0.7
D	0.1	0.2	0.7	11

(c) normalizing error rates

	A	B	C	D
A	1	0.7	0.2	0.1
B	0.7	1	0.1	0.2
C	0.2	0.1	1	0.7
D	0.1	0.2	0.7	1

(d) filling diagonals

Table 5: Confusion matrices showing the outcome after each step of preprocessing from the example data.

3.1 Data: Perception errors

We analyzed perception errors from Table III (shown in Table 8 in the Appendix) of Miller and Nicely (1955), which summarizes the result from a syllable identification experiment. In the experiment, the stimuli are [Cɑ] with 16 English consonants as the onset. The acoustic stimuli underwent frequency modulation, and noise was added to the stimuli. The data from the condition with the widest-band noise (200-6500 Hz) was chosen in the current study. This choice was made to avoid potential biases due to the exclusion of frequency ranges of greater importance for a subset of features. The condition with a relatively low S/N ratio of -12 dB was chosen so that weakly similar phonemes could still be confused with each other, potentially revealing more information about features that are usually well preserved during transmission.

3.2 Data: Production errors

Speech error data were collected from the Fromkin Speech Error Database web interface.[1] The database contains spontaneous speech errors from natural speech. The search query included "English" as the "target language," "phonological" as the "error type," "substitution" as the "process procedure," and "all" in other fields. The entries that also had "addition" or "exchange" as the "process procedure" in any analysis were excluded. Then, entries were manually removed if they involved the following: (1) a change in the number of segments in the same syllable component (e.g., "small" →

[1] https://www.mpi.nl/dbmpi/sedb/sperco_form4.pl

"fall"; [ɜɹ] was considered a single segment); (2) changes of multiple syllable components (e.g. "detectors" →"locators"); (3) blending of two words (e.g., "jumped"/ "leapt" →[dʒipt] "jeapt"); (4) mispronunciation due to orthography (e.g., [sɑm] "psalm" →[pɑm] "palm"). Only phonemes that were present in both production and perception data were kept in the analysis, namely, the sixteen consonants [p t k b d g f v θ ð s z ʃ ʒ m n]. This resulted in 455 production errors summarized in Table 9.

3.3 Evaluation

To evaluate how well the extracted features correspond to the features that are actually used in the English language, we examine the the feature sets' capacities in defining natural classes, which are the groups of phonemes that pattern together in phonological alternations.

English rule-based sound patterns from P-Base (Mielke, 2008) were used to extract natural classes in English phonology. The English patterns in P-Base were produced with reference to Jensen (1993); McMahon (2002). The search resulted in 9 rule-based natural classes (found as the left environment, the right environment, the target, or the output of the rule). Some natural classes contain phonemes that are not included in the 16 consonants for feature extraction in this study—in these cases, the extra phonemes were removed. The patterns yielded 9 unique natural classes.

The evaluation of a discovered feature set was based on that feature set's minimal feature definition for the set of phonemes that is the closest to attested natural class in terms of the number of different phonemes, where the feature definition is formed by a single feature value or by the conjunction of multiple feature values.

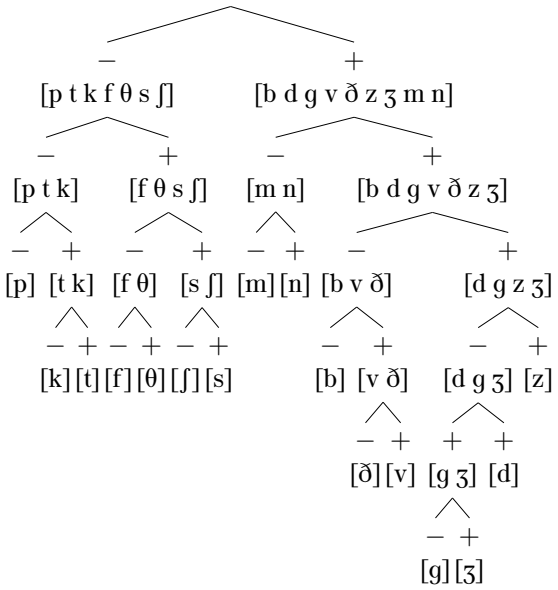
We also tested how well a reference set of distinctive features could define the natural classes to compare with extracted feature sets. We use a set of seven phonological features from *the Sound Pattern of English* (SPE; Chomsky and Halle (1968)). We take these features to be reasonably well adapted to capturing English phonological classes, and thus a useful point of contact with English phonology. The SPE features included are [nasal] ([nas]), [voice] ([voi]), [continuant] ([cont]), [strident] ([strid]), [coronal] ([cor]), [anterior] ([ant]), and [distributed].[2]

[2]Since [distributed] is underspecified for velars, in the class definition test, velars are considered as [-distributed] to make the [distributed] feature comparable to other features.

3.4 Results

Here we present the extracted results and compare the extracted features with traditional phonological features. The goal of this section is to assess whether the discovered features are meaningful beyond describing the errors in perception/production.

3.4.1 Perception



1: voi	[p t k f θ s ʃ | b d g v z ð ʒ m n]
2: nas cont	[p t k m n | b d g f θ s ʃ v z ð ʒ]
3: pl$_1$	[p b f θ v ð m | t k d g s ʃ z ʒ n]
4: cont pl$_2$	[p k b d g f ʃ ʒ m | t θ s v z ð n]
5	[p k g θ s z ð ʒ m | t b d f ʃ v n]
6	[p k b d g θ s v n | t f ʃ z ð ʒ m]

Figure 1: The binary feature set extracted from the perception data, presented as a tree (above) and as lists of phonemes split by the "| " symbol (below). For the sake of visual presentation, we leave nodes that do not branch off of the tree, but it should be noted that the features are fully specified: all phonemes have some value for every feature.

As shown in figure 1, the first extracted perception feature accurately differentiates the voiced phonemes from the voiceless phonemes. The second perception feature divides the two sub-clusters created by the first feature based on two different properties. Among voiceless sounds, it divides fricatives from plosives. Meanwhile, among voiced sounds, it creates a division based on nasality. We remark that, unlike the hierarchical clustering methods alluded to in the introduction, which perform a myopic subdivision of each cluster—ignoring

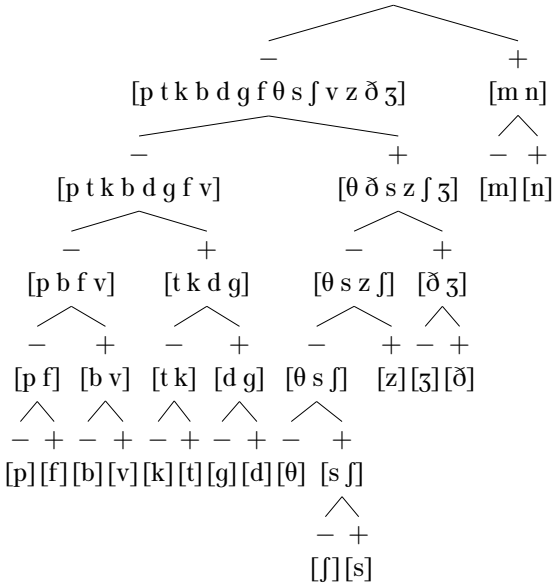
all of the phonemes outside it—the algorithm we employ here only ever discovers features that are specified for every phoneme in the inventory. It is therefore curious that, in this example, we see an apparently myopic behaviour, whereby the second discovered feature picks out a (physically) different phonetic property depending on the value of the first discovered feature. In addition to the fact that the perception data may capture patterns that would not be obvious from an objective phonetic point of view, it should be underscored that, while the algorithm's use of fully-specified features means that it *can* capture commonalities that cross-cut the whole inventory, nothing *requires* that these commonalities be the decisive factor in selecting a feature. In this case, it is difficult to determine whether the attribution of a common feature value to nasals and voiceless plosives is perceptually meaningful or whether it is merely an artefact of the algorithm's need to construct fully-specified features.

The third feature groups the labial and interdental consonants against the consonants that are further back. We will explore this "[front]" feature further below. The rest of the extracted features complete the other divisions needed to distinguish all phonemes, but do not clearly correspond to phonological properties.

3.4.2 Production

As shown in figure 2, the first production feature corresponds to nasality. In the non-nasal subset that the first feature induces, the second feature mostly corresponds to the [cont] feature, with the exception that the labiodental fricatives [f v] are grouped with the stops. This pattern might suggest an intermediate status for English labiodental consonants between fricatives and stops. Just like in the perception-based features, the behaviour of the second feature is different for the nasal versus the non-nasal subset: it divides the two nasals by place of articulation.

The third feature also picks out phonetically different classes depending on the featurally-defined subset. Among the stops, it separates labial sounds from coronal and velar sounds. Among the fricatives, however, it separates [ð ʒ] from the rest. The fourth feature corresponds to [voice] with the exception of [ʒ m], which are both grouped with voiceless segments. The fifth feature mostly contrasts coronal against non-coronal sounds; in the clusters where there are only labial sounds, it separates the sounds based on continuancy. The last feature provides the last remaining contrast between [ʃ s].



1: nas	[p t k b d g f θ s ʃ v z ð ʒ | m n]
2: cont	[p t k b d g f v m | θ s ʃ z ð ʒ n]
3: pl$_1$ voi	[p b f θ s ʃ v z m | t k d g ð ʒ n]
4: voi	[p t k f θ s ʃ ʒ m | b d g v z ð n]
5: pl$_2$ cont	[p k b g θ ð m | t d f s ʃ v z ʒ n]
6	[p k b g ʃ z ʒ m | t d f θ s v ð n]

Figure 2: The binary feature set extracted from the speech error data, presented as a tree (above) and as lists of phonemes split by the "\|" symbol (below). For the sake of visual presentation, we leave nodes that do not branch off of the tree, but it should be noted that the features are fully specified: all phonemes have some value for every feature.

3.4.3 Defining natural classes

The performance in defining natural classes is summarized in Table 6. Recall that, for each natural class in the list of English natural classes, we seek to find the conjunction of features that gives the most similar set of phonemes.

The first column indicates how many of the natural classes allow an exact match. We see that the SPE feature set is the most capable in defining natural classes, followed by the production feature set, while the perception feature set performs the worst. There is one of the natural classes [p t k f θ] that the SPE feature set cannot define. This class includes all voiceless obstruents except for [s ʃ]. In fact, this class, which appears in P-Base, is apparently the result of an overly surface-oriented characterization of an English phonological pat-

Features	Classes successfully captured	Mean minimal feature number for matches
production	6	2.5
perception	4	2.8
SPE	8	2

Table 6: Defining natural classes (n=9) in English rule-based patterns with different feature sets by feature conjunction.

tern: it is that set of consonants for which, if they are at the end of a noun, a plural suffix would be realized as [s] (rather than [z] or [əz]). This alternation in the plural suffix is usually described with two phonological rules (devoicing and epenthesis), rather than with reference to this superficial class. The two classes required in the underlying rules are voiceless consonants and sibilants, which can both be defined by the SPE features. The performance is better when this class is excluded—and we note that none of the discovered feature sets can characterize it either. The second column shows the average number of features required to define the exact-matched natural classes. Again, the SPE feature set does best, followed by the production and then the perception features.

Here we discuss the definitions of two example classes. The first class is the interdental consonants [θ ð]. This class showcases that the same group of consonants may be captured differently by three feature sets. SPE defines it with [+continuant,-strident]. The perception feature set defines it with [+2,-3,-4] (**+2** is [b d g f θ s ʃ v z ð ʒ], **-3** is [p b f θ v ð m], **-4** is [p k g θ s z ð ʒ m]). The production feature set defines it with two features [+2,-5] (**+2** is [θ s ʃ z ð ʒ n]; **-5** is [p k b g θ ð m]). Note that neither of the two extracted feature sets utilizes features that only target fricatives like the SPE feature [strident].

The second class, alveolar obstruents [t d s z], shows the limit of the extracted feature sets. It can be defined by the SPE features [+coronal -nasal -distributed]. But both production and perception feature sets failed to accurately define this class: the closest sets defined by the two feature sets are [t d f s ʃ v z ʒ] and [t k d g s ʃ z ʒ n], respectively.

4 Discussion

4.1 Algorithm

As discussed above, the algorithm may "meld" features across sub-inventories: for example, the second feature discovered from the production data divides obstruents by continuancy, but divides nasals by place. The nature of the redundancy term contributes to this problem. An alternative feature encoding only continuancy would not split the nasals at all. As this would lead to greater similarity to the previous feature (which also groups the nasals together), this is dispreferred by the redundancy term. One potential future direction for automatic feature extraction method is to develop a criterion for assigning the weight of the redundancy term so that this tendency could be controlled.

4.2 Data sets

In the production data set, errors were collected by multiple linguists in daily conversations. This might introduce biases. First, the phonemes are not equally distributed in natural speech. This contributes to the lack of errors related to the phonemes [ð ʒ]. Second, because the speech error data is based on researchers' perception of speech, it is inherently influenced by the biases in perception (Alderete and Davies, 2019; Pouplier and Goldstein, 2005), for examples, researchers might have different criteria for correct pronunciation and might miss some errors that are more difficult to hear. The Fromkin Speech Error Database is the most suitable publicly available English production data for feature extraction at the time of this study. However, researchers have started collecting new data sets with more systematic approaches to address these issues, for example, the Simon Fraser University Speech Error Database Cantonese 1.0 (Alderete, 2023). Applying our feature extraction algorithm to these new data sets could potentially reveal more accurate featural information in production.

In the perception data set, the errors were collected from the identification of noise-masked syllable audio. The design of the noise could impact different features unequally, which also might introduce biases in feature extraction.

Together, these observations point to a deeper question: if the goal of inferring features from data is to arrive at a single, common representation, how might multiple, sometimes contradictory, types of data be productively combined into a single analy-

sis? The commonalities between the two learned feature sets above are promising—the presence of features encoding nasality, voicing, and continuancy in both—but also highlight important differences: voicing is more prominent in the perception data, while nasality is more prominent in the production data. These kinds of inconsistencies may pose challenges for combining data sets.

4.3 Insights into English consonant features

As discussed above, the English labiodental consonants behave similarly to plosives in production error data, and, as a result, share a feature in the analysis. The consequences of such a move for the analysis of English are not immediately obvious, but the idea that these phonemes have an intermediate continuancy status has not previously be considered to our knowledge.

Second, considering the extracted features from both production and perception errors, a set of two potential place features are suggested in Table 7.

	[+front]	[−front]
[+peripheral]	[b p (f v) m]	[k g ʃ ʒ]
[−peripheral]	[(f v) θ ð]	[t d s z n]

Table 7: A possible four-way place distinction for 16 English phonemes. [f v] may be specified as either [+peripheral] or [−peripheral].

The suggested [front] feature is supported by the third perception feature and the resembling third production feature. This [front] feature is similar to the [anterior] SPE feature, the difference between the two being the membership of the alveolar consonants.

The [peripheral] feature in this system is based on the fifth production feature and a similar feature that is the fourth perception feature. It is similar to the *Peripheral* constituent proposed by Rice (1994). The difference is that Rice's *Peripheral* constituent only encompasses the features *Labial* and *Dorsal*, while the feature [peripheral] here also includes the fricatives [ʃ ʒ]. Besides the similarity with *Peripheral*, if the labiodental fricatives [f v] are analyzed as [+peripheral], then the [−peripheral] feature would also be the same as the [dental] feature of SPE (Chomsky and Halle, 1968).

5 Summary of contributions

The current study is the first-ever attempt to extract cross-classifying features, as opposed to mere classes, from phoneme confusion data in perception and production. The extracted feature sets from two modalities differ, but both show links to phonological properties. Familiar features such as voicing, nasality, and continuancy are seen in both extracted feature sets. The extracted feature sets also showed interesting deviations from commonly used phonological features, including the different features based on the frontness and peripherality of consonants. These alternative extracted features are also useful in defining natural classes, with the production features having a better performance, showing more connection between phonology and production errors than the connection between phonology and perception errors.

6 Data availability

Code and data is available at https://github.com/zhanaofu/speech-feature-extraction.

7 Acknowledgments

Supported by the Connaught Fund and the Arts and Science Bridging Fund, U. of Toronto, Natural Sciences and Engineering Research Council of Canada (NSERC) RGPIN-2022-04431 and RGPIN-2017-06053. The authors are grateful to Jeff Mielke for his help in the project.

References

John Alderete. 2023. Cross-linguistic trends in speech errors: An analysis of sub-lexical errors in Cantonese. *Language and Speech*, 66(1):79–104.

John Alderete and Monica Davies. 2019. Investigating Perceptual Biases, Data Reliability, and Data Discovery in a Methodology for Collecting Speech Errors From Audio Recordings. *Language and Speech*, 62(2):281–317.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row, New York.

Victoria A Fromkin. 1971. The Non-Anomalous Nature of Anomalous Utterances. *Language*, 47(1):27–52.

John T Jensen. 1993. *English Phonology*, volume 99. John Benjamins Publishing.

Ying Lin. 2005. *Learning Features and Segments from Waveforms: A Statistical Model of Early Phonological Acquisition*. University of California, Los Angeles.

Ying Lin and Jeff Mielke. 2006. Discovering place and manner features—What can be learned from acoustic

and articulatory data? *The Journal of the Acoustical Society of America*, 120(5):3136–3136.

April McMahon. 2002. *An Introduction to English Phonology*. Edinburgh University Press.

Jeff Mielke. 2008. *The Emergence of Distinctive Features*. Oxford University Press.

Jeff Mielke. 2012. A phonetically based metric of sound similarity. *Lingua*, 122(2):145–163.

George A. Miller and Patricia E. Nicely. 1955. An Analysis of Perceptual Confusions Among Some English Consonants. *The Journal of the Acoustical Society of America*, 27(2):338–352.

Marianne Pouplier and Louis Goldstein. 2005. Asymmetries in the perception of speech production errors. *Journal of Phonetics*, 33(1):47–75.

Keren Rice. 1994. Peripheral in Consonants. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 39(3):191–216.

Cory Shain and Micha Elsner. 2019. Measuring the perceptual availability of phonological features during language acquisition using unsupervised binary stochastic autoencoders. In *Proceedings of the 2019 Conference of the North*, pages 69–85, Minneapolis, Minnesota. Association for Computational Linguistics.

Nikolai Sergeevich Trubetzkoy. 1969. *Principles of Phonology*. University of California Press, Berkeley.

Michael S Vitevitch. 2002. The influence of phonological similarity neighborhoods on speech production. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(4):735.

A Appendix

Perceived phoneme	Phoneme in audio: p	t	k	f	θ	s	ʃ	b	d	g	v	ð	z	ʒ	m	n
p	80	43	64	17	14	6	2	1	1		1	1			2	
t	71	84	55	5	9	3	8	1				1	2		2	3
k	66	76	107	12	8	9	4					1			1	
f	18	12	9	175	48	11	1	7	2	1	2	2				
θ	19	17	16	104	64	32	7	5	4	5	6	4	5			
s	8	5	4	23	39	107	45	4	2	3	1	1	3	2		1
ʃ	1	6	3	4	6	29	195		3							1
b	1			5	4	4		136	10	9	47	16	6	1	5	4
d							8	5	80	45	11	20	20	26	1	
g					2			3	63	66	3	19	37	56		3
v				2		2		48	5	5	145	45	12		4	
ð					6			31	6	17	86	58	21	5	6	4
z					1	1	1	7	20	27	16	28	94	44		1
ʒ								1	26	18	3	8	45	129		2
m	1							4			4	1	3		177	46
n					4			1	5	2		7	1	6	47	163

Table 8: Perception errors from Table III in Miller and Nicely (1955).

Pronounced phoneme	Intended phoneme: p	t	k	b	d	g	f	θ	s	ʃ	v	ð	z	ʒ	m	n
p		12	15	8	1		12		7						4	1
t	8		7		3		1	3	6						3	1
k	15	8		4	5	4	3		5						1	1
b	7	3	3		6	3	7				4				10	
d	1	6	3	4		3			5		1		1		1	5
g			8	5	5										1	
f	15	4	2	5	1			4	8		10				2	
θ	1	3							4							
s	1	4	4	1	3		7	7		3			2			
ʃ		2	1	1			1	1	31			1		2		
v			2	3			8	1	1				5			
ð					1			1			1					
z									4		2					1
ʒ										1			1			
m	9			6		1	4		3	1			1			9
n		9			3				1						15	

Table 9: Single-phoneme substitution production errors extracted from the Fromkin Speech Error Database.

	p	t	k	b	d	g	f	θ	s	ʃ	v	ð	z	ʒ	m	n
[nas]	−	−	−	−	−	−	−	−	−	−	−	−	−	−	+	+
[voi]	−	−	−	+	+	+	−	−	−	−	+	+	+	+	+	+
[cont]	−	−	−	−	−	−	+	+	+	+	+	+	+	+	−	−
[strid]	−	−	−	−	−	−	+	−	+	+	+	−	+	+	−	−
[cor]	−	+	−	−	+	−	−	+	+	+	−	+	+	+	−	+
[ant]	+	+	−	+	+	−	+	+	+	−	+	+	+	−	+	+
[dist]	+	−		+	−		−	+	−	+	−	+	−	+	+	−

Table 10: SPE features (Chomsky and Halle, 1968)

Filtering input for learning constrained grammatical variability: The case of Spanish word order

Shalinee Maitra
Department of Linguistics
University of California Los Angeles
Los Angeles, CA 90095-1543
shalinee30@ucla.edu

Laurel Perkins
Department of Linguistics
University of California Los Angeles
Los Angeles, CA 90095-1543
perkinsl@ucla.edu

Abstract

Children learn basic word order from data in which both subjects and objects can appear in variable positions. Spanish learners acquire a word order that deterministically places objects after verbs, and allows variation only in subject position. We present a model for acquiring this type of constrained variability from messy data. Our model expects that (1) its data contain a mixture of signal and noise for canonical word order, and (2) subjects control agreement on verbs. We find that this model can learn to filter noise from its data to identify the canonical word order for Spanish while a model that does not track subject-verb agreement cannot. These results suggest that having expectations about the types of regularities that the data will contain can help learners identify variability that is constrained along certain dimensions.

1 Introduction

Children acquire the canonical word order of their language at young ages, from input that contains a mixture of canonical and non-canonical word orders whose structure they cannot yet represent (Hirsh-Pasek and Golinkoff, 1996; Perkins and Lidz, 2021, 2020). Non-canonical sentences like *wh*-questions introduce perceived variability into learners' data, which they must abstract away from in order to identify basic subject and object position. However, some types of variability are part of the core grammatical phenomenon to be acquired. In Spanish, full lexical objects canonically must occur after verbs, but subject position is not fully deterministic: subjects can occur both pre- and post-verbally in basic clauses (1-2) (Lozano, 2006; Domínguez and Arche, 2008; De Prada Pérez and Pascual y Cabo, 2012). Learners must identify that this variability is a property of the language's basic clause syntax, whereas other variability is due to subject or object displacement in non-canonical sentence types (3). How do learners identify that basic subject position varies, but object position is fixed, if both argument positions appear to be variable in their data?

(1) *Mariela tiró la pelota.* (basic SVO)
Mariela throw-PAST-SG the-SG ball-SG
'Mariela threw the ball.'

(2) *Entró Mariela.* (basic VS)
Enter-PAST-SG Mariela
'Mariela entered.'

(3) *¿Cuál pelota tiró Mariela?* (*wh*-Q, OVS)
Which-SG ball-SG throw-PAST-SG Mariela
Which ball did Mariela throw?

On one proposal, learners might avoid being misled by messy data by assuming that some portion of their data is "noise," introduced by grammatical processes they cannot yet account for. Successful learning arises when learners are able to infer which portion of their data to treat as noise, and which portion to treat as signal for the rules governing the phenomenon they are trying to acquire (Perkins and Hunter, 2023; Perkins et al., 2022; Schneider et al., 2020). This can be seen as a mechanism for "regularization" in learning (Hudson Kam and Newport, 2005, 2009; Culbertson et al., 2013) whereby learners acquire a system that allows less variability than the data that they are learning from. But the case of Spanish word order poses a challenge for this approach. Here, learners must abstract away from certain *types* of variability— for instance, the noise introduced by non-canonical sentence types— while treating other types of variability as informative about the phenomenon to be acquired. That is, learners must identify that they should "regularize" along only certain dimensions.

We propose that learners might solve this problem by using knowledge about the specific types of regularities that grammars tend to exhibit. In the case of word order acquisition, learners might

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 108-120.
Amherst, Massachusetts, June 15-17, 2023

expect that subjects and objects will enter into different sorts of grammatical dependencies— for instance, that subjects tend to control agreement on verbs. We present a learner that looks for evidence of subject-verb agreement in its data, and uses this information to infer which portion of its data to treat as signal for underlying basic word order. We show that this learner is able to identify constrained variation in Spanish word order. We also show that our learner performs substantially better than a learner that does not track subject-verb agreement. This suggests that for certain types of grammatical generalizations, successful learning requires knowledge of the sorts of dependencies that grammars make available, along with mechanisms for detecting relevant evidence in noisy data.

2 Acquiring word order in Spanish

Cross-linguistically, children learn basic word order in infancy (Perkins and Lidz, 2020; Hirsh-Pasek and Golinkoff, 1996; Franck et al., 2013; Gavarró et al., 2015; Zhu et al., 2022). They do so at ages even before they have adult-like representations for non-canonical clause types where this basic word order is distorted. For instance, infants learning English identify that their language is canonically SVO even before they can identify that arguments have been moved in *wh*-questions (Hirsh-Pasek and Golinkoff, 1996; Perkins and Lidz, 2021). This suggests that learners have a way to implicitly "filter" the messiness introduced by non-canonical clause types when learning basic clause syntax (Pinker, 1984; Gleitman, 1990; Lidz and Gleitman, 2004).

On one proposal, learners might infer how to separate "signal" for the grammatical phenomenon being acquired from "noise" introduced by various other processes (Perkins et al., 2022; Perkins and Hunter, 2023). This inference is possible even if learners do not know ahead of time which of the utterances they hear should be treated as noise—for instance, because they have not yet learned what basic vs. non-basic clauses look like. Perkins and Hunter (2023) show that a learner can use the distributions of imperfectly-identified noun phrases and verbs in child-directed speech to determine which data to treat as signal for basic word order, without prior expectations about where noise will occur. Their model successfully filtered its noisy input in order to infer that French and English have canonical SVO word order. A similar mechanism has been applied to model the successful acquisition of verb transitivity classes (Perkins et al., 2022).

Here, we ask whether this same type of filtering mechanism can succeed in cases of more variable word order. In Spanish, full lexical objects are obligatorily postverbal, but subjects can occur both before and after the verb in basic clauses.[1] But a variety of constructions obscure evidence for these basic word orders. For instance, *wh*-dependencies and topic and focus constructions introduce frequent argument displacement. Furthermore, Spanish has frequent null subjects, which cause a unique ambiguity for learning basic word order. For a child at early stages of syntactic development, sentences like (4) and (5) may be structurally ambiguous. If the child does not know the meaning of these words and whether null subjects are present, it is unclear whether the noun phrase after the verb is the subject or the object.

(4) *Traen los regalos.*
pro bring-PL the-PL gift-PL
'(They) bring the gifts.'

(5) *Llegan los profesores.*
arrive-PL the-PL teacher-PL
'The teachers arrive.'

On the basis of ambiguous data like (4) and (5), we can imagine at least two erroneous conclusions that the learner may reach. On the one hand, the learner might conclude that both of these sentences are transitive with null subjects, making the postverbal noun phrases both objects. This would mean that the learner is missing relevant evidence for postverbal subjects in the language. On the other hand, the learner might decide that both of these sentences are intransitive, and the postverbal noun phrases are both subjects. This would mean that the learner is missing relevant evidence for postverbal objects in the language. If this type of data is prevalent, the learner may need additional information to draw the correct conclusion that the language has both postverbal subjects and postverbal objects.

One possible source of information that could help children reach the correct conclusion is subject-verb agreement. Because objects do not agree with verbs while subjects do, postverbal nominals do not always match verbs in number (6). This

[1] In basic clauses with broad focus, postverbal subjects typically occur in intransitive clauses with unaccusative rather than unergative verbs (De Prada Pérez and Pascual y Cabo, 2012). There is also debate regarding the canonical clausal position of subjects in Spanish (Villa-García, 2012). We abstract away from these issues in the current discussion.

agreement asymmetry reflects a cross-linguistic tendency: in languages where verbs agree with an argument, that argument is typically a subject (Moravcsik, 1974, 1978; Gilligan, 1987).[2]

(6) *Trae los regalos.*
pro bring-SG the-PL gift-PL
(He) brings the gifts.

If children expect subjects to control agreement on verbs, and can find evidence for these agreement dependencies in their data, then number mismatches like the one in (6) could help them identify the postverbal noun phrase as an object and not a subject. Furthermore, a proliferation of postverbal noun phrases that agree with verbs could provide evidence for postverbal subjects, particularly if these occur at a rate higher than would be expected if they were all objects.

In languages that morphologically mark subject-verb agreement, there is evidence that infants can track these patterns from very young ages (Nazzi et al., 2011), along with other types of morphologically-marked dependencies (Van Heugten and Shi, 2010; Soderstrom et al., 2007; Hohle et al., 2006; Santelmann and Jusczyk, 1998). It is not clear how abstractly children represent these types of dependencies at young ages (Culbertson et al., 2016), but these sensitivities make it plausible that they might use them in the process of word order acquisition, particularly in a language like Spanish that has rich and transparent agreement morphology.

Can a filtering mechanism of the sort proposed in previous literature successfully acquire the constrained variability in Spanish word order, given the range of noise in the data that children will encounter? We present a model that learns from strings of imperfectly-represented noun phrases and verbs. It learns to filter noise from its data in order to identify canonical word order, using evidence for subject-verb number agreement but no further cues to sentence structure. We find that the learner is able to successfully identify that Spanish has postverbal objects and variation in subject position. Moreover, this learner performs substantially better than a learner that relies on the distributions of noun phrases and verbs alone, without expecting subjects and verbs to agree. Thus, solving this problem may require not only the ability to learn in a noise-tolerant way from distributions in data, but also expectations about the types of agreement dependencies that clause arguments enter into.

[2]Some languages mark object as well as subject agreement, while others do not mark subject verb agreement. Two relevant questions for future work are (i) how a learner would identify multiple agreement dependencies in languages with more complex agreement systems and (ii) how a learner would fare in a language with fewer agreement dependencies.

3 Model

We adapt a Bayesian learner from Perkins and Hunter (2023). The model observes strings of noun phrases and verbs tagged for number features. The model assumes that its observed strings have been generated by some mixture of canonical and non-canonical grammatical processes. Specifically, the learner chooses among discrete composite probabilistic context-free grammars (PCFGs) that contain different sets of "core" rules governing canonical word order (e.g., SVO, SOV, etc.), and a shared set of "noise" rules that introduce additional variability into the data. We compare two models whose hypothesis spaces contain different sets of composite PCFGs, one that expects subject-verb number agreement ('Agreement Model') and one that does not ('No-Agreement Model'). The model seeks to divide its data into signal and noise in order to identify which combination of core and noise rules best explains the distributions it observes.

3.1 Generative Model

The grammars in the *Agreement Model* generate strings with exactly one verb, either singular or plural (v-sg or v-pl), and up to two noun phrases, either singular or plural (np-sg or np-pl). Two of the grammars in the Agreement Model's hypothesis space are shown in Table 1: one whose canonical word order is SVO, and one whose canonical word order requires objects to occur after verbs but allows subjects to vary in their position ('VO', the target word order of Spanish). In these grammars, NP-pl is deterministically rewritten as np-pl, NP-sg as np-sg, V-pl as v-pl, and V-sg as v-sg; these are not shown for purposes of space.

These grammars enforce subject-verb argreement in their core rules by requiring, for S expansions, that only an NP-pl occurs with a VP-pl and only an NP-sg occurs with a VP-sg. However, for VP expansions, both NP-pl and NP-sg are allowed to occur with a V-sg or V-pl, so verbs are not required to agree with direct objects in number.

The learner chooses among nine possible grammars of this sort, whose core rules correspond to

SVO Core Rules	VO Core Rules	Shared Noise Rules	
S → NP-pl VP-pl	S → NP-pl VP-pl	S ⇢ NP-pl VP-pl	S ⇢ VP-pl
S → NP-sg VP-sg	S → NP-sg VP-sg	S ⇢ NP-sg VP-sg	S ⇢ VP-sg
	S → VP-pl NP-pl	S ⇢ VP-pl NP-pl	
	S → VP-sg NP-sg	S ⇢ VP-sg NP-sg	
VP-pl → V-pl NP-pl	VP-pl → V-pl NP-pl	VP-pl ⇢ V-pl NP-pl	VP-pl ⇢ NP-pl V-pl
VP-pl → V-pl NP-sg	VP-pl → V-pl NP-sg	VP-pl ⇢ V-pl NP-sg	VP-pl ⇢ NP-sg V-pl
VP-pl → V-pl	VP-pl → V-pl	VP-pl ⇢ V-pl	
VP-sg → V-sg NP-pl	VP-sg → V-sg NP-pl	VP-sg ⇢ V-sg NP-pl	VP-sg ⇢ NP-pl V-sg
VP-sg → V-sg NP-sg	VP-sg → V-sg NP-sg	VP-sg ⇢ V-sg NP-sg	VP-sg ⇢ NP-sg V-sg
VP-sg → V-sg	VP-sg → V-sg	VP-sg ⇢ V-sg	

Table 1: SVO and VO grammars, Agreement Model

nine distinct word order options. We model the learning process as a choice among these nine discrete grammars; see Perkins and Hunter (2023) for discussion of the role of discreteness in the learner's hypothesis space in this type of model. These grammars include the four most restricted word orders, where subjects deterministically occur either before or after the verb phrase and objects before or after the verb: SVO, SOV, OVS, and VOS (the four options arising from a 2x2 choice of subject and object position). The hypothesis space also includes a 'Free' word order that allows any ordering of subjects and objects, and four word orders that allow some degree of variation: two that fix object position as either OV or VO and allow subjects on either side of the verb phrase; and two that fix subject position as either SV or VS and allow objects on either side of the verb. Note that each of these last four grammars essentially combine two of the more restricted grammars. In particular, the VO grammar (the target word order for Spanish) is the union of the VOS and SVO grammars. See the Appendix for full details.

In addition to the core rules that generate canonical word order, each grammar has a set of noise rules (represented by dashed arrows in Table 1) that manipulate the same set of terminal and nonterminal symbols as the core rules, but allow for all possible permutations and deletions of clause arguments. Each of the nine grammars in the learner's hypothesis space has the same set of noise rules. This allows all of the grammars to generate any of the strings in the dataset. For example, the SVO grammar can generate the string v-pl np-sg np-pl via the trees in Fig. 1. In the first tree, two noise rules are used: the noisy S expansion places the subject after the VP, and the noisy VP expansion places the object after the verb. Notice that it is also possible for a tree to be generated by a mixture of

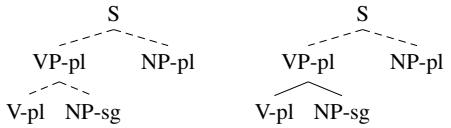


Figure 1: Two possible analyses of v-pl np-sg np-pl (suppressing NP-sg → np-sg, NP-pl → np-pl and V-pl → v-pl rewrites) where solid lines indicate core rules and dashed lines indicate noise rules

core and noise rules, as in the second tree: here, the S expansion is noisy, but the VP can be expanded according to the core rules of the SVO grammar.

The core rules of these grammars do not contain the rules S → VP-sg and S → VP-pl, meaning that the learner expects canonical clauses to have subjects. These expansions of S only occur in the noise rules; subject-drop is assumed to be a process that introduces noise for basic word order learning.

The *No-Agreement Model* is just like the Agreement Model, except that the grammars in its hypothesis space do not encode subject-verb number agreement. These grammars generate strings that contain exactly one v and up to two nps, not marked for number. The SVO grammar and the VO grammar are shown in Table 2. In these grammars, NP is deterministically rewritten as np and V is deterministically rewritten as v; these are again omitted for the sake of space.

The No-Agreement Model has the same nine word order options as the Agreement Model in its hypothesis space: the four most deterministic word orders, four that allow variability in either subject or object position, and one that allows both subject and object position to vary. Each of these nine grammars again shares the same set of noise rules, which allow any word ordering as well as argument deletion. Just as in the Agreement Model, subjectless clauses are only allowed via the grammars' noise rules.

SVO Core Rules	VO Core Rules	Shared Noise Rules
S → NP VP	S → NP VP	S ⇢ NP VP
	S → VP NP	S ⇢ VP NP
		S ⇢ VP
VP → V NP	VP → V NP	VP ⇢ V NP
VP → V	VP → V	VP ⇢ V
		VP ⇢ NP V

Table 2: SVO and VO grammars, No-Agreement Model

Agreement	No Agreement
0.38 v-sg	0.5 v
0.18 v-sg np-sg	0.25 v np
0.14 v-pl	0.12 np v
0.08 np-sg v-sg	0.06 np v np
0.04 np-sg v-sg np-sg	0.05 v np np
0.04 v-pl np-sg	0.02 np np v
0.03 v-sg np-sg np-sg	
0.02 v-pl np-pl	
0.02 np-sg v-pl	

Table 3: Proportions of most frequent string types

For both models, the prior distribution over the nine grammars G in the learner's hypothesis space is uniform, meaning each of the nine grammars has the same prior probability. This means that none of the canonical word orders is preferred *a priori*. Each of the allowable core and noise rules in these grammars has some probability associated with it. To work with these rule probabilities, we recast the composite grammars illustrated in Tables 1 and 2 into standard PCFGs, following Perkins and Hunter (2023).[3] For every nonterminal N in these grammars, we add additional nonterminals N^+ and N^-. The expansions for N^+ and N^- are determined by the grammar's core and noise rules, respectively. We also add the rules $N \rightarrow N^+$ and $N \rightarrow N^-$, whose weights represent the probabilities for using a core vs. noise expansion of N. Let $\vec{\theta}_{n_G}$ be the vector of probabilities for expanding a nonterminal n in the resulting standard PCFG for G. The prior distribution over $\vec{\theta}_{n_G}$ is a Dirichlet distribution with parameters α_{n_G}. We set all components of α in these distributions to 1, which results in a uniform distribution over the rule probabilities. This means that all core expansions of a given nonterminal are equally likely *a priori*, as are all noise expansions.

Each grammar conditions a distribution over trees and strings. Just as for any standard PCFG, the probability of generating a string via a particular tree under grammar G is the product of the rule probabilities $\vec{\theta}_G$ used in that tree. To calculate the overall probability of a string under grammar G, we sum over the probabilities of all of the ways that it could be generated.

3.2 Inference

Our model infers the posterior probability distribution over its grammars G and an approximation of trees $\vec{t}$ given its observed strings $\vec{w}$. Following Perkins and Hunter (2023), instead of inferring a distribution over $\vec{t}$ directly, we sample approximations of trees, which we call 'coarse structures', $\vec{s}$. These coarse structures abstract away from the core vs. noise distinctions in the trees. For example, both trees in Fig. 1 would share the same coarse structure: the same tree without a distinction between dashed and solid lines. Abstracting away from this distinction means that all grammars in the learner's hypothesis space can generate every coarse structure, using either noise rules, or core rules, or some combination. This allows for feasible sampling of grammars given a sample of coarse structures.

We use Gibbs Sampling to estimate the joint posterior probability of grammars and coarse structures, $P(G, \vec{s} \mid \vec{w})$, summing over all combinations of core and noise options in $\vec{s}$ and integrating over $\vec{\theta}$. The steps of sampling work as follows. First, G is randomly initialized to one of the nine grammars in the hypothesis space. Then, we alternate between drawing samples from the posterior probability of a grammar given a set of coarse structures for the observed strings, $P(G \mid \vec{w}, \vec{s})$, and the posterior probability of coarse structures given a grammar and the observed strings, $P(\vec{s} \mid \vec{w}, G)$.

Via Bayes' Rule, the posterior probability of a grammar given coarse structures and strings, $P(G \mid \vec{w}, \vec{s})$, is proportional to the likelihood of the strings and coarse structures given the grammar, times the prior probability of that grammar:

$$(1) \quad P(G \mid \vec{w}, \vec{s}) = \frac{P(\vec{s}, \vec{w} \mid G)P(G)}{\sum_{G'} P(\vec{s}, \vec{w} \mid G')P(G')}$$

We assume that all grammars have equal prior probability, and calculate the likelihood $P(\vec{s}, \vec{w} \mid G)$ following Perkins and Hunter (2023). After sampling a new grammar from the posterior distribution in Eq. (1), we sample a new set of coarse structures from $P(\vec{s} \mid \vec{w}, G)$ using a Hastings pro-

[3]This formalization bears resemblance to a latent variable PCFG (Cohen, 2017), in which the choice between noise (−) vs. non-noise (+) at each nonterminal node could be recast as a choice of a particular latent state. We thank an anonymous reviewer for pointing this out.

posal, following a method introduced in Johnson et al. (2007). These steps are repeated until the chain converges to a stable distribution which estimates the joint posterior $P(G, \vec{s} \mid \vec{w})$. We refer the reader to Perkins and Hunter (2023) for more details of the sampling procedure.

For the results reported below, 20,000 iterations of Gibbs Sampling were performed. Every tenth sample of the last 10,000 iterations was analyzed.

4 Simulations

4.1 Data

We tested our learners on datasets of child-directed Spanish created from the Fernandez/Aguado corpus in CHILDES (Fernandez Vazquez and Gerardo Aguado). The corpus includes a total of 45,610 utterances directed to 47 different children between the ages of approximately 3;0 and 4;0. This corpus was chosen because of its large size and the large number of children included, allowing for more reliable estimates of the distributions that any given child might hear.

The dataset for the Agreement Model consisted of strings of noun phrases and verbs annotated with number features. We conducted an automatic search of the corpus, using a heuristic that aimed to approximate the immature grammatical category knowledge of an infant learning basic word order. Because young infants can differentiate nouns from verbs using determiners, auxiliaries, and pronouns (Babineau and Christophe, 2022; Shi and Melançon, 2010; Hicks et al., 2007) we noisily identified noun phrases and verbs in the corpus using these functional cues. All full pronouns were included as np's, with their number determined by the form of the pronoun. Any word occurring after a determiner was counted as the head of an np, and its number was determined based on the inflection of the determiner. Any word occurring after an auxiliary was counted as a v, and its number was determined by the inflection on the auxiliary. Proper names were counted as np-sg's. *Wh*-words and clitics were not counted as np's, because there is no evidence that children identify these as nominals before learning basic word order (Perkins and Lidz, 2021; Brusini et al., 2017).

After these strings were extracted, only strings with exactly one verb and up to two noun phrases where at least one noun phrase matched the verb in number were retained. From this subset of the corpus, we calculated the proportion of each string type, and sampled 25 strings according to these proportions. This resulted in 9 string types included in the dataset for the Agreement learner (see Table 3). This dataset is substantially noisy: nearly 60% of these strings cannot not be generated by the core rules of the VO grammar, which is the target word order of Spanish, without using noise rules.

The dataset for the No-Agreement learner was generated by the same process and heuristics for finding noun phrases and verbs, but number features were not tagged.[4] We sampled 25 strings according to their proportions in the corpus, resulting in the 6 string types in Table 3. Just as in the dataset for the Agreement Model, almost 60% of these strings cannot be generated by the core rules of the VO grammar without the option of noise.

4.2 Results

Figure 2 shows the posterior distribution over grammars inferred by the Agreement and the No-Agreement Model, averaged across 10 runs of each learner. In these graphs, the dashed line represents no substantial learning: a learner that maintains its prior belief that all of its 9 grammars are equally probable would infer a distribution with all bars hovering around 0.11.

The No-Agreement Model inferred roughly this flat distribution. The target VO grammar, along with most other grammars, was assigned posterior probability around 0.11. VOS and OVS were assigned slightly higher posterior probability (both a mean of 0.14); overall, the model gave slightly higher probability to the more restrictive grammars. The fact that all grammars were assigned low and approximately equal probability suggests that the No-Agreement Model did not learn much useful information about Spanish word order.

The Agreement Model, by contrast, inferred a substantially different distribution. Three of the learner's grammars received much higher probability than the other six. These three grammars are VO (mean posterior probability: 0.23), SVO (mean: 0.20), and VOS (mean: 0.28). The other grammars

[4]There are certain strings that were present in the No Agreement dataset that were not present in the Agreement dataset. For example, the string np-sg v-pl np-sg would not be included in the Agreement dataset because the Agreement grammars cannot generate this string (since neither np agrees with the verb in number), but this string would be tagged as np v np under the heuristics for the No Agreement dataset, and thus would be included. This is why the proportions in the Agreement dataset in Table 3 do not add up to the relevant proportions in the No Agreement dataset.

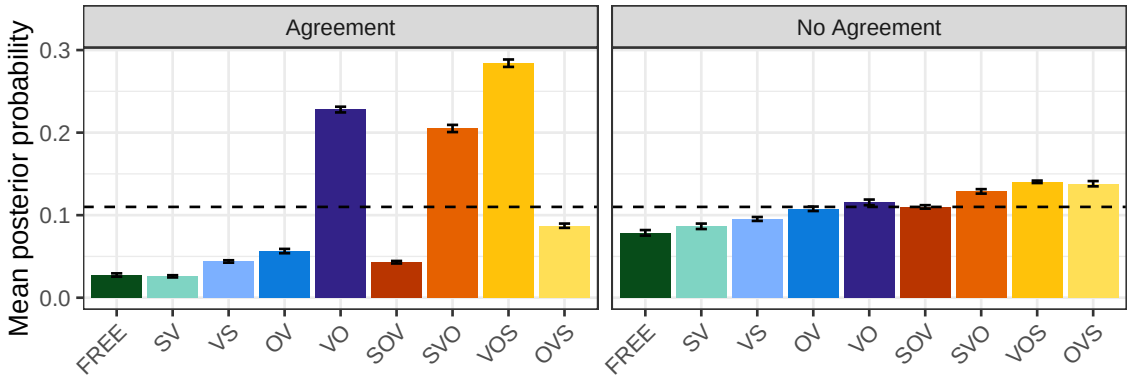


Figure 2: Posterior distribution over word-order grammars (G)

were all assigned much lower probability, ranging from 0.03 (SV) to 0.09 (OVS).

While the target VO grammar is among the three that the learner identified as having highest posterior probability, it did not identify this grammar as the single most probable. However, looking more closely at these results, we see that the learner's inference was fairly sensible. All three grammars with stand-out posterior probability only allow postverbal objects, which indicates that the learner successfully identified Spanish object position. Furthermore, the strings that the target VO grammar can generate are exactly the combination of the strings generated by the SVO and VOS grammars. So, the fact that these three grammars were assigned the highest posterior probability indicates that the learner had success in determining that object position is fixed, but subject position varies. This inference is striking given the degree of noise that the learner needed to overcome: nearly 60% of the strings in its data were not consistent with the canonical word order options that it successfully identified, without taking noise into account.

Interestingly, we see that the learner's inferred distribution favors VOS by a small amount. Why would this be the case? One reason may be that this type of Bayesian learner prefers more restrictive hypotheses. This is a phenomenon known as "Bayesian Occam's Razor," under which the hypothesis with the fewest degrees of freedom that can explain all the data will be preferred (Griffiths et al., 2008). In the case of these models, the SVO and VOS grammars correspond to hypotheses with fewer degrees of freedom than the more flexible VO grammar. Spanish allows both of these word orders, so the combination of explaining the data well and having fewer degrees of freedom gives VOS a small advantage over VO, and gives SVO high probability as well. The same preference for restrictive hypotheses is visible in the No Agreement Model, where the four most constrained grammars tended to receive higher posterior probability than the more flexible ones.

The learner's slight preference for VOS points to an additional limitation in its search for subject-verb agreement. The strings that provide the best evidence for VOS are the v-initial strings in which there is at least one postverbal np that matches the v in number: our learner will tend to take this match as evidence for subject-verb agreement, and analyze these strings as having postverbal subjects. These strings make up 23% of the learner's dataset, lending support to grammars in which the subject is fixed postverbally. However, because Spanish allows null subjects, a number of these postverbal np's are likely to be objects rather than subjects: this is the ambiguity demonstrated in (4-5) in Section 2. If a child were only tracking number agreement, like our learner, perhaps that child would likewise mis-analyze many of these sentences.

Possible extensions of this learner might leverage other information in order to overcome this bias towards VOS. One of the potential cues that is available in Spanish, but is removed by our preprocessing of the data, is person agreement. Tracking person features would give the learner an additional way to disambiguate between subjects and objects. Of the v-initial strings in which the v and a potential subject np match in number, approximately 25% *mismatch* in person features (see Table 4). These person mismatches could help a more sophisticated learner identify that many of these strings are underlyingly verb-object, not verb-subject, just as mismatches in number features can disambiguate

V-initial string type	Prop. person mismatches
v-sg np-sg np-sg	0.28
v-sg np-sg	0.25
v-pl np-pl	0.16
Overall	**0.25**

Table 4: Person mismatches in relevant v-initial strings

these parses in cases like (6). An example is shown in (7), where the 3rd-person postverbal object *a ella* mismatches the 1st-person inflected verb *veo*.

(7) *La veo a ella.*
pro her-3SG see-1SG to her-3SG
'(I) see her.'

So, a learner that makes use of a wider range of evidence for subject-verb agreement might overcome its bias towards determinism, and infer with higher probability that subject position is variable.

In sum, our results show that the Agreement Model was able to use its expectation of subject-verb agreement to abstract away from a great degree of noise in its data and infer a canonical word order in which objects are obligatorily postverbal, with some variation in subject position. By contrast, the No-Agreement Model failed to infer that any of its hypothesized canonical word orders were more probable than any of the others. Thus, tracking subject-verb number agreement helped substantially in this learning problem. A learner that expected subjects to agree with verbs was able to draw reasonable inferences about Spanish word order on the basis of noisy data; a learner with no awareness of agreement could not.

5 Discussion

We present a model for learning constrained variability in Spanish word order. Spanish learners need to acquire a word order with obligatorily postverbal objects and variable subject position from messy data, in which both subjects and objects might appear to vary in position. We extend an approach introduced by Perkins and Hunter (2023) to model this learning as a case of separating "signal" for basic word order from "noise" from non-canonical clause types. We pursue the hypothesis that, in solving this problem, learners may make use of knowledge that subjects and verbs will tend to agree. We compare a learner that attempts to identify a grammar of canonical word order using subject-verb number agreement to a learner that relies entirely on noun phrase and verb distributions. We find that the model that tracks subject-verb agreement is able to infer Spanish word order, whereas the model with no knowledge of agreement cannot. This suggests that knowledge of the types of dependencies that clause arguments enter into may helpfully guide word order learning.

Our case study demonstrates how tolerant this learning mechanism is to noise: the learner succeeds at identifying the target canonical word order even though approximately 60% of the data appears inconsistent with that order. The learner's noise-tolerance comes in part from its ability to find useful information in sub-parts of strings, instead of treating each string as either entirely signal or entirely noise. The learner assumes that noise can occur in any of the internal nodes in a tree individually, so it entertains the possibility that a string could be generated with a mixture of core vs. noise rules, as shown in Figure 1. This allows the learner to look within strings to find evidence for the grammatical regularities it expects, thereby making use of more of its data.

Thus, if Spanish-learning children are reliably able to track subject-verb agreement at the age when they are learning word order, then they might be able to use agreement to aid in this task, even in the absence of other reliable cues to sentence structure (e.g., from meaning or prosody; Pinker, 1984; Christophe et al., 2008). However, this depends on children knowing the morphological forms of number and potentially person inflection in the language. Prior work shows that French learners track subject-verb dependencies in infancy (Nazzi et al., 2011), and learners in various languages track similar dependencies at young ages (Van Heugten and Shi, 2010; Soderstrom et al., 2007; Hohle et al., 2006; Santelmann and Jusczyk, 1998). However, we do not know precisely when children begin to track these dependencies, and how reliably and abstractly they represent them (Culbertson et al., 2016). Further work could explore whether our filtering mechanism would succeed even if learners have noisy or incomplete representations of these dependency types. These findings also invite further behavioral work on the acquisition of agreement in Spanish and similar languages.

Our model provides a window into the mechanisms for acquiring basic clause syntax in a language with frequent argument-drop and complex argument realization patterns. Subject pro-drop is a frequent and basic property of Spanish; how-

ever, our model treats this as a type of noise to ignore, and expects that canonical clauses will have overt subjects. While learners must eventually acquire pro-drop in Spanish, it may make sense for a learner to only attempt to learn canonical subject position from overt arguments, setting aside subject-drop as a phenomenon to be acquired independently. Indeed, in exploratory simulations, we find that allowing null subjects in the learner's core grammar rules does not help it identify Spanish word order; what helps is knowledge of subject-verb agreement. Our model therefore makes the prediction that knowledge of subject-verb agreement, but not necessarily pro-drop, may need to be acquired prior to the acquisition of word order in Spanish— a prediction that could be tested in future behavioral work. Beyond Spanish, many languages with argument-drop and more variable word orders also have rich case and agreement systems. The model presented here could therefore be extended to explore how case and agreement dependencies may inform learning in languages with diverse argument structure profiles.

These results have broader implications for our understanding of when and how learners regularize variable data (Hudson Kam and Newport, 2005, 2009; Reali and Griffiths, 2009; Ferdinand et al., 2019). We highlight a distinction between forms of regularization in which learners (i) abstract away from variability in data in order to draw fully deterministic generalizations, and (ii) draw generalizations that are not fully deterministic, but are still more constrained than the data would appear to support. For the current case study, we propose that learners use knowledge about the kinds of regularities that grammars tend to exhibit in order to identify which types of variability they should learn from, and which types they should treat as noise. This mechanism may generalize to other areas in language acquisition and learning in other domains, in which learners' regularization tendencies arise from the expectation that their data will noisily reflect a richly structured underlying system.

Acknowledgments

We thank Xinyue Cui, Tim Hunter, Victoria Mateu, the UCLA Psycholinguistics/Computational Linguistics Seminar, and three anonymous reviewers for helpful feedback and assistance.

References

Mireille Babineau and Anne Christophe. 2022. Preverbal infants' sensitivity to grammatical dependencies. *Infancy*, 27(4):648–662.

Perrine Brusini, Ghislaine Dehaene-Lambertz, Marieke Van Heugten, Alex De Carvalho, François Goffinet, Anne-Caroline Fiévet, and Anne Christophe. 2017. Ambiguous function words do not prevent 18-month-olds from building accurate syntactic category expectations: An erp study. *Neuropsychologia*, 98:4–12.

Anne Christophe, Séverine Millotte, Savita Bernal, and Jeffrey Lidz. 2008. Bootstrapping Lexical and Syntactic Acquisition. *Language and Speech*, 51(1-2):61–75.

Shay B Cohen. 2017. Latent-variable pcfgs: Background and applications. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 47–58.

Jennifer Culbertson, Elena Koulaguina, Nayeli Gonzalez-Gomez, Géraldine Legendre, and Thierry Nazzi. 2016. Developing knowledge of nonadjacent dependencies. *Developmental Psychology*, 52(12):2174.

Jennifer Culbertson, Paul Smolensky, and Colin Wilson. 2013. Cognitive biases, linguistic universals, and constraint-based grammar learning. *Topics in Cognitive Science*, 5(3):392–424.

Ana De Prada Pérez and Diego Pascual y Cabo. 2012. Interface heritage speech across proficiencies: unaccusativity, focus, and subject position in spanish. In *Selected Proceedings of the 14th Hispanic Linguistics Symposium*, pages 308–318. Cascadilla Proceedings Project Somerville, MA.

Laura Domínguez and María J Arche. 2008. Optionality in L2 grammars: the acquisition of SV/VS contrast in Spanish. In *Proceedings of the 32nd Annual Boston University Conference on Language Development*, Cambridge, MA. Cascadilla Press.

Vanessa Ferdinand, Simon Kirby, and Kenny Smith. 2019. The cognitive roots of regularization in language. *Cognition*, 184:53–68.

Marta Fernandez Vazquez and Alonso Gerardo Aguado. CHILDES Database Spanish Fernandez/Aguado Corpus.

Julie Franck, Severine Millotte, Andres Posada, and Luigi Rizzi. 2013. Abstract knowledge of word order by 19 months: An eye-tracking study. *Applied Psycholinguistics*, 34(2):323–336.

Anna Gavarró, Maya Leela, Luigi Rizzi, and Julie Franck. 2015. Knowledge of the OV parameter setting at 19 months: Evidence from Hindi–Urdu. *Lingua*, 154:27–34.

Gary Martin Gilligan. 1987. *A cross-linguistic approach to the pro-drop parameter.* Ph.D. thesis, University of Southern California, Los Angeles, CA.

Lila Gleitman. 1990. The structural sources of verb meanings. *Language Acquisition*, 1(1):3–55.

Thomas L Griffiths, Charles Kemp, and Joshua B Tenenbaum. 2008. Bayesian models of cognition. In R. Sun, editor, *The Cambridge handbook of computational psychology*, page 59–100. Cambridge University Press.

J Hicks, J Maye, and J Lidz. 2007. The role of function words in infants' syntactic categorization of novel words. In *Proceedings of the Linguistic Society of America Annual Meeting*, Anaheim, CA.

Kathy Hirsh-Pasek and Roberta Michnick Golinkoff. 1996. The intermodal preferential looking paradigm: A window onto emerging language comprehension. In Dana McDaniel, Cecile McKee, and Helen S. Cairns, editors, *Methods for assessing children's syntax*, pages 105–124. The MIT Press, Cambridge, MA.

Barbara Hohle, Michaela Schmitz, Lynn M Santelmann, and Jurgen Weissenborn. 2006. The recognition of discontinuous verbal dependencies by german 19-month-olds: Evidence for lexical and structural influences on children's early processing capacities. *Language Learning and Development*, 2(4):277–300.

Carla L Hudson Kam and Elissa L Newport. 2005. Regularizing unpredictable variation: The roles of adult and child learners in language formation and change. *Language Learning and Development*, 1(2):151–195.

Carla L Hudson Kam and Elissa L Newport. 2009. Getting it right by getting it wrong: When learners change languages. *Cognitive Psychology*, 59(1):30–66.

Mark Johnson, Thomas L Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov Chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146.

Jeffrey Lidz and Lila R Gleitman. 2004. Argument structure and the child's contribution to language learning. *Trends in Cognitive Sciences*, 8(4):157–161.

Cristóbal Lozano. 2006. Focus and split-intransitivity: the acquisition of word order alternations in non-native spanish. *Second Language Research*, 22(2):145–187.

Edith Moravcsik. 1974. Object-verb agreement. *Working Papers on Language Universals*, 15:25–140.

Edith A Moravcsik. 1978. Agreement. In Joseph H Greenberg, Charles A Ferguson, and Edith A Moravcsik, editors, *Universals of Human Language. IV: Syntax*, pages 331–74. Stanford University Press, Stanford.

Thierry Nazzi, Isabelle Barrière, Louise Goyet, Sarah Kresh, and Géraldine Legendre. 2011. Tracking irregular morphophonological dependencies in natural language: Evidence from the acquisition of subject-verb agreement in french. *Cognition*, 120(1):119–135.

Laurel Perkins, Naomi H Feldman, and Jeffrey Lidz. 2022. The power of ignoring: filtering input for argument structure acquisition. *Cognitive Science*, 46(1):e13080.

Laurel Perkins and Tim Hunter. 2023. Noise-tolerant learning as selection among deterministic grammatical hypotheses. In *Proceedings of the 6th meeting of the Society for Computation in Linguistics (SCiL 2023)*.

Laurel Perkins and Jeffrey Lidz. 2020. Filler-gap dependency comprehension at 15 months: The role of vocabulary. *Language Acquisition*, 27(1):98–115.

Laurel Perkins and Jeffrey Lidz. 2021. Eighteen-month-old infants represent nonlocal syntactic dependencies. *Proceedings of the National Academy of Sciences*, 118(41):e2026469118.

Steven Pinker. 1984. *Language learnability and language development.* Harvard University Press.

Florencia Reali and Thomas L Griffiths. 2009. The evolution of frequency distributions: Relating regularization to inductive biases through iterated learning. *Cognition*, 111(3):317–328.

Lynn M Santelmann and Peter W Jusczyk. 1998. Sensitivity to discontinuous dependencies in language learners: Evidence for limitations in processing space. *Cognition*, 69(2):105–134.

Jordan Schneider, Laurel Perkins, and Naomi H Feldman. 2020. A noisy channel model for systematizing unpredictable input variation. In *Proceedings of the 44th Annual Boston University Conference on Language Development*, pages 533–547.

Rushen Shi and Andréane Melançon. 2010. Syntactic categorization in french-learning infants. *Infancy*, 15(5):517–533.

Melanie Soderstrom, Katherine S White, Erin Conwell, and James L Morgan. 2007. Receptive grammatical knowledge of familiar content words and inflection in 16-month-olds. *Infancy*, 12(1):1–29.

Marieke Van Heugten and Rushen Shi. 2010. Infants' sensitivity to non-adjacent dependencies across phonological phrase boundaries. *The Journal of the Acoustical Society of America*, 128(5):EL223–EL228.

Julio Villa-García. 2012. Spanish subjects can be subjects: Acquisitional and empirical evidence. *Iberia: An International Journal of Theoretical Linguistics, 4 (2), 124-169.*

Jingtao Zhu, Julie Franck, Luigi Rizzi, and Anna Gavarró. 2022. Do infants have abstract grammatical knowledge of word order at 17 months? evidence from Mandarin Chinese. *Journal of Child Language*, 49(1):60–79.

A Complete List of Grammars

VO Core Rules	**OV Core Rules**	**SV Core Rules**	**VS Core Rules**	**Free Core Rules**
S → NP-pl VP-pl	S → NP-pl VP-pl	S → NP-pl VP-pl	S → VP-pl NP-pl	S → NP-pl VP-pl
S → NP-sg VP-sg	S → NP-sg VP-sg	S → NP-sg VP-sg	S → VP-sg NP-sg	S → VP-pl NP-pl
S → VP-pl NP-pl	S → VP-pl NP-pl			S → NP-sg VP-sg
S → VP-sg NP-sg	S → VP-sg NP-sg			S → VP-sg NP-sg
VP-pl → V-pl NP-pl	VP-pl → NP-pl V-pl	VP-pl → NP-pl V-pl	VP-pl → NP-pl V-pl	VP-pl → NP-pl V-pl
VP-pl → V-pl NP-sg	VP-pl → NP-sg V-pl	VP-pl → NP-sg V-pl	VP-pl → NP-sg V-pl	VP-pl → NP-sg V-pl
		VP-pl → V-pl NP-pl	VP-pl → V-pl NP-pl	VP-pl → V-pl NP-pl
		VP-pl → V-pl NP-sg	VP-pl → V-pl NP-sg	VP-pl → V-pl NP-sg
VP-pl → V-pl	VP-pl → V-pl	VP-pl → V-pl	VP-pl → V-pl	VP-pl → V-pl
VP-sg → V-sg NP-sg	VP-sg → NP-pl V-sg	VP-sg → NP-pl V-sg	VP-sg → NP-pl V-sg	VP-sg → NP-pl V-sg
VP-sg → V-sg NP-pl	VP-sg → NP-sg V-sg	VP-sg → NP-sg V-sg	VP-sg → NP-sg V-sg	VP-sg → NP-sg V-sg
		VP-sg → V-sg NP-pl	VP-sg → V-sg NP-pl	VP-sg → V-sg NP-pl
		VP-sg → V-sg NP-sg	VP-sg → V-sg NP-sg	VP-sg → V-sg NP-sg
VP-sg → V-sg	VP-sg → V-sg	VP-sg → V-sg	VP-sg → V-sg	VP-sg → V-sg

SVO Core Rules	**SOV Core Rules**	**VOS Core Rules**	**OVS Core Rules**
S → NP-pl VP-pl	S → NP-pl VP-pl	S → VP-pl NP-pl	S → VP-pl NP-pl
S → NP-sg VP-sg	S → NP-sg VP-sg	S → VP-sg NP-sg	S → VP-sg NP-sg
VP-pl → V-pl NP-pl	VP-pl → NP-pl V-pl	VP-pl → V-pl NP-pl	VP-pl → NP-pl V-pl
VP-pl → V-pl NP-sg	VP-pl → NP-sg V-pl	VP-pl → V-pl NP-sg	VP-pl → NP-sg V-pl
VP-pl → V-pl	VP-pl → V-pl	VP-pl → V-pl	VP-pl → V-pl
VP-sg → V-sg NP-pl	VP-sg → NP-pl V-sg	VP-sg → V-sg NP-pl	VP-sg → NP-pl V-sg
VP-sg → V-sg NP-sg	VP-sg → NP-sg V-sg	VP-sg → V-sg NP-sg	VP-sg → NP-sg V-sg
VP-sg → V-sg	VP-sg → V-sg	VP-sg → V-sg	VP-sg → V-sg

Shared Noise Rules	**Shared Terminal Rules**
S ⇢ NP-pl VP-pl	NP-pl → np-pl
S ⇢ VP-pl NP-pl	NP-sg → np-sg
S ⇢ VP-pl	V-pl → v-pl
S ⇢ NP-sg VP-sg	V-sg → v-sg
S ⇢ VP-sg NP-sg	
S ⇢ VP-sg	
VP-pl ⇢ NP-pl V-pl	
VP-pl ⇢ NP-sg V-pl	
VP-pl ⇢ V-pl NP-pl	
VP-pl ⇢ V-pl NP-sg	
VP-pl ⇢ V-pl	
VP-sg ⇢ NP-pl V-sg	
VP-sg ⇢ NP-sg V-sg	
VP-sg ⇢ V-sg NP-pl	
VP-sg ⇢ V-sg NP-sg	
VP-sg ⇢ V-sg	

Table 5: All Agreement Grammars

VO Core Rules	**OV Core Rules**	**SV Core Rules**	**VS Core Rules**	**Free Core Rules**
S $\rightarrow$ NP VP	S $\rightarrow$ NP VP	S $\rightarrow$ NP VP	S $\rightarrow$ VP NP	S $\rightarrow$ NP VP
S $\rightarrow$ VP NP	S $\rightarrow$ VP NP			S $\rightarrow$ VP NP
VP $\rightarrow$ V NP	VP $\rightarrow$ NP V	VP $\rightarrow$ NP V	VP $\rightarrow$ NP V	VP $\rightarrow$ NP V
		VP $\rightarrow$ V NP	VP $\rightarrow$ V NP	VP $\rightarrow$ V NP
VP $\rightarrow$ V	VP $\rightarrow$ V	VP $\rightarrow$ V	VP $\rightarrow$ V	VP $\rightarrow$ V

SVO Core Rules	**SOV Core Rules**	**VOS Core Rules**	**OVS Core Rules**
S $\rightarrow$ NP VP	S $\rightarrow$ NP VP	S $\rightarrow$ VP NP	S $\rightarrow$ VP NP
VP $\rightarrow$ V NP	VP $\rightarrow$ NP V	VP $\rightarrow$ V NP	VP $\rightarrow$ NP V
VP $\rightarrow$ V	VP $\rightarrow$ V	VP $\rightarrow$ V	VP $\rightarrow$ V

Shared Noise Rules	**Shared Terminal Rules**
S $\dashrightarrow$ NP VP	NP $\rightarrow$ np
S $\dashrightarrow$ VP NP	V $\rightarrow$ v
S $\dashrightarrow$ VP	
VP $\dashrightarrow$ NP V	
VP $\dashrightarrow$ V NP	
VP $\dashrightarrow$ V	

Table 6: All No-Agreement Grammars

An incremental RSA model for adjective ordering preferences in referential visual context

Fabian Schlotterbeck and **Hening Wang**
University of Tübingen
Tübingen, Germany
{fabian.schlotterbeck,hening.wang}@uni-tuebingen.de

Abstract

We report data from a preference rating experiment that tested for conflicting effects of subjectivity and discriminatory strength on adjective ordering preferences in referential visual context. Results indicate that, if the communicative efficiency of an adjective is low in a given context, it is preferred later in a multi-adjective expression. To account for qualitative aspects of these data, we propose a novel computational model of incremental processing in the Rational Speech Act framework. What sets the model apart from previous approaches is that it assumes fully incremental interpretation, without the need to anticipate possible sentence completions.

1 Introduction

In noun phrases (NPs) with multiple adjectives, as in (1), the relative order of the adjectives can vary, but at the same time, there are robust cross-linguistic preferences (Sproat and Shih, 1991) such that certain adjective sequences are more common and perceived as more natural than others. For example, the ordering in (1-a) is strongly preferred to that in (1-b).

(1) a. big white bear
b. white big bear

Although adjective ordering preferences have been known and studied for some time, they have resisted a unified explanation. Existing explanations come from different perspectives in linguistics and include semantic hierarchies (Dixon, 1982), syntactic mapping (Cinque, 1993) and psycholinguistic explanations based on as absoluteness (Martin, 1969) or closeness to the meaning of head noun (Whorf, 1945). Here, we focus on two recent hypotheses (Scontras et al., 2017; Fukumura, 2018, see next section for explanation) that have gained support from experimental work and share a common theoretical motivation. In particular, they are both based on the idea that efficiency in communication determines ordering preferences. Despite being based on the same general idea, theses hypotheses may lead us to expect significantly divergent outcomes in certain contexts. To address this tension, we pit these predictions against each other in a preference rating experiment. Furthermore, we implement both hypotheses in a novel computational model of incremental interpretation in the Rational Speech Act (RSA, Frank and Goodman, 2012) framework that not only provides a qualitative explanation of our findings but also sheds light on the relative contribution of the two hypotheses.

2 Two rational explanations of adjective ordering

The first explanation we focus on was proposed by Scontras et al. (2017), who showed that the subjectivity of adjectives is a strong predictor of ordering preferences. We call this the SUBJECTIVITY hypothesis. They operationalized subjectivity as *faultless disagreement*, roughly the degree to which two speakers can disagree about attributing a property to an individual without one of them necessarily being wrong. According to the SUBJECTIVITY hypothesis, (1-a) is preferred over (1-b) because *big* is more subjective than *white* and is also further away from the noun. In fact, gradable dimension adjectives like *big*, *tall* or *heavy* are prime examples of subjective adjectives that have received a lot of attention in previous work. We therefore focus the following discussion on these instances. In subsequent work, Scontras et al. (2019) proposed that the low communicative efficiency of subjective expressions is one possible reason for effects of subjectivity on ordering preferences. The main idea of Scontras et al. (2019) is that more efficient expressions are integrated earlier in the hierarchical structure underlying semantic composition in order to minimize the risk of misidentification of referents, and thus, as a result, these expressions

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 121-132.
Amherst, Massachusetts, June 15-17, 2023

end up closer to the modified noun in the linear sequence (at least in languages with prenominal modification).

The SUBJECTIVITY hypothesis has gained support from corpus studies as well as preference rating experiments in a variety of languages (Scontras et al., 2020b). Furthermore, the idea that communicative efficiency is increased if the more subjective expressions enter later into compositional meaning derivations was corroborated in computational simulations of rational communication (Simonic, 2018; Franke et al., 2019, see section 5 for discussion).

Another explanation of ordering preferences was given by Fukumura (2018), who investigated the impact of the *discriminatory strength* of adjectives. In a given context, a referring expression has greater discriminatory strength if it contains more information about the intended referent. If it singles out the intended referent perfectly, it has maximal strength. The main idea of the DISCRIMINATORY STRENGTH hypothesis is that the more discriminatory an adjective is, the more salient and accessible it will be in a visual context and also the more useful for reference resolution. Consequently, there will be a higher likelihood of early mention in the linear sequence (and thus greater distance from the noun in prenominal modification).

Fukumura (2018) tested the DISCRIMINATORY STRENGTH hypothesis in a production experiment where participants described referents that were marked in visual context. Discriminatory strength was controlled by manipulating the properties of the presented objects. In addition, color adjectives were compared to adjectives describing patterns, e.g. *striped*. As expected based on previous studies, Fukumura (2018) found that color adjectives were preferred before pattern adjectives and she explained this by a high availability of color adjectives in production. In addition, she found that discriminatory strength had the predicted effect and higher discriminatory strength in context led to earlier mention in the participants' productions. However, since there is no strong subjectivity gradient between color and pattern adjectives, her results do not speak to the SUBJECTIVITY hypothesis and the question remains open how these two hypotheses are related to each other.

3 Relation between SUBJECTIVITY and DISCRIMINATORY STRENGTH

Both the SUBJECTIVITY and the DISCRIMINATORY STRENGTH hypothesis are based on the idea that ordering preferences emerge from pressures towards efficient communication and both of them assume that more informative expressions are in some sense used "**earlier**". However, the two hypotheses take different perspectives and thus arrive at different definitions of what "early" means. In particular, the SUBJECTIVITY hypothesis is derived from the perspective of a listener whereas DISCRIMINATORY STRENGTH assumes a speaker perspective. The listener aims to identify an intended referent by sequentially restricting a set of potential referents in a process that follows the compositional semantic structure of a given expression. Thus, the listener evaluates the adjective that is closer to the noun first (thereby interpreting (1-a) as referring to bears that are big for white bears). As a consequence, the hierarchical structure of the NP determines what counts as "early" in the SUBJECTIVITY hypothesis. The speaker, by contrast, aims to maximize informativity at each step in the word-by-word production of an utterance. In the DISCRIMINATORY STRENGTH hypothesis, the position in the linear sequence of words is thus central. For these reasons, "earlier" translates to either **closer to the noun** or **further away from the noun**, depending on which perspective we take.

This is, in fact, a striking difference between the SUBJECTIVITY and the DISCRIMINATORY STRENGTH hypothesis and it is an interesting empirical question what happens if these two perspectives stand in direct conflict to each other. This could, e.g., be the case in a context in which a less subjective adjective discriminates more strongly than a more subjective one between the intended referent and a set of distractors. This exact question is the main question we addressed in the experiment reported in the next section, in which participants indicated their preferences between multi-adjective expressions like in (1) when referring to a target referent in visual context.

To appreciate the purpose and limitations of our experiment, it may be worthwhile to reflect briefly on the predictions that can be derived from the SUBJECTIVITY hypothesis in the type of contextually-embedded experimental setting underlying the DISCRIMINATORY STRENGTH hypothesis of Fukumura (2018). We acknowledge that, strictly speak-

ing, the SUBJECTIVITY hypothesis, by itself, does not predict how preferences are affected by manipulations of visual context. This is because SUBJECTIVITY does not presuppose that subjective-first expressions are less informative in every setting. There only need to be enough such instances overall for a general preference to "evolv[e] gradually" (Franke et al., 2019; cf. also Scontras, 2023). Thus, the SUBJECTIVITY hypothesis explicitly allows for counterexamples. One such counterexample is the case where a multi-adjective expression like in (1) receives a conjunctive instead of the assumed "sequentially intersective" reading (cf. Franke et al., 2019), such that (1-a) would be understood as referring to bears that are white and big (for bears) rather than big for white bears. In fact, Scontras et al. (2020a) presented empirical evidence that the preference for subjective-first orderings vanishes when adjectives restrict the set of potential referents in conjunction. We cannot exclude the possibility that the specific design of our current experiment constitutes another counterexample, maybe even because conjunctive readings are favored in our design. Be this as it may, a gradual evolution of the SUBJECTIVITY-based preferences that are commonly observed would be extremely challenging to explain based on low informativity of subjective expressions if we find empirically that speakers actually adapt by producing subjective adjectives more often in first position (in the linear sequence) if context renders them more (rather than less) informative.

4 Experimental Data: Preference ratings in visual contexts

4.1 Method

In a web-based experiment, we collected data on adjective ordering preferences in German using preference ratings of multiple adjective sequences in visual referential context. Participants (N=120) were recruited via the platform *prolific.co*. They were instructed at the begin of the experiment by a cover story that they should communicate a target sticker (marked with a red box, see Fig. 1) in a scrapbook to an imagined listener on a telephone call. With this setting, we aimed to rule out the possibilities of using information of relative spatial positions in the context and tried to simulate an online communication situation as closely as possible. In each experimental trial, participants were presented with a visual context and they indicated their preference between two sentences with reversed adjective order using a slider in the middle of the screen (see Fig. 1).

In a mixed factorial design, we manipulated, within participants, the COMBINATION of adjectives from different semantic classes (levels: *dimension & either color or shape* and *color & shape*) and the RELEVANCE of the corresponding properties for reference resolution, i.e. whether the *first*, *second* or *both* properties were needed to identify a referent (cf. Fig. 1). [1] The purpose of these two factors was to test whether the basic findings of Fukumura (2018) replicate also with subjective adjectives and, in particular, whether the preference for subjective adjectives in first position persists if the more subjective adjective has the lesser discriminatory strength.

In addition to this within-participants manipulation, we also manipulated the SIZE DISTRIBUTION of objects (*sharp* vs. *blurred*) between-participants. As in Fig. 1, there were always six objects in the visual context that were either large or small. The large objects had sizes that were randomly sampled from the integers 9 and 10 (in some arbitrary unit of length that effectively depended on the display settings of the experimental participants). If size was the relevant property, the target object was always the biggest, irrespective of SIZE DISTRIBUTIONS. In *sharp* SIZE DISTRIBUTION, sizes of the remaining, small objects were sampled from the integers in the range $[1, 3]$ whereas they were sampled from $[1, 6]$ in *blurred* SIZE DISTRIBUTION. As a result, the small objects in the *blurred* as compared to the *sharp* distribution had greater variance in size among them and a smaller mean distance to the sizes of the big objects. The idea behind this manipulation was to affect the information that size adjectives could convey in such a way that they are more useful in *sharp* vs. *blurred* distributions. In particular, we intended to make size adjectives effectively non-subjective in *sharp* distributions. If any prediction about the effect of this manipulation can be derived from the SUBJECTIVITY hypothesis (see discussion above), the preference

[1] The factor COMBINATION was originally a three-level factor with the levels *dimension & color*, *dimension & shape* and *color & shape*. We aggregated the first two levels here because they did not differ significantly and their distinction is not relevant for our present purpose, in particular for the computational models described in section 6. The complete design and statistical analysis along with a free production experiment in the same general design is described in the unpublished MA thesis of Wang (2022).

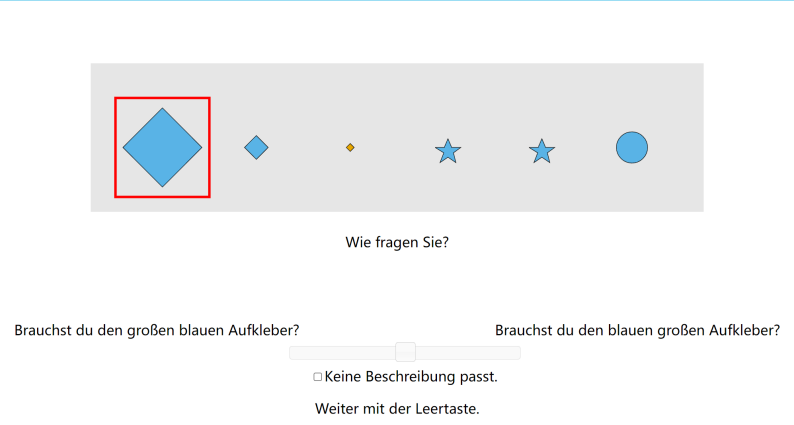


Figure 1: An example item from the current experiment in the condition with COMBINATION of *dimension and color* adjectives and RELEVANCE of the *first* property (i.e. *dimension*) in a *sharp* SIZE DISTRIBUTION. A property was counted as relevant if it was necessary for referent identification. In this example size is relevant but color and shape are not. Glosses for the German linguistic material in the example item are provided in Appendix A.

for subjective-first orders should therefore be weakened in *sharp* SIZE DISTRIBUTION. The reason is that the SUBJECTIVITY hypothesis assumes that less subjective adjectives are integrated earlier into the hierarchical structure.

We generated 27 experimental items in each of the 18 conditions, resulting in a total of 486 items that were distributed across 6 lists (three per SIZE DISTRIBUTION). Each participant saw a total of 81 experimental items. These were combined with 99 filler items that were constructed in a similar way as the experimental items but also included sentences with only one adjective instead of two. Overall, each participant thus completed 180 trials. An experimental session took around half an hour and participants received reimbursement of 5.25 £.

4.2 Results

The mean slider positions are shown in Fig 2. For statistical analysis, we used linear mixed effects models (Bates et al., 2015) that incorporated fixed effects of all manipulated factors and their interactions, along with random intercepts for participants and items. For hypothesis testing, we used model comparisons based on log-likelihood ratio tests. First of all, our results replicate effects of SUBJECTIVITY: There was a strong preference for dimension adjectives in first position which resulted in a significant effect of COMBINATION on slider ratings ($\chi^2(1) = 361.97, p < .001$). Furthermore, we found a significant interaction between RELEVANCE and SIZE DISTRIBUTION ($\chi^2(2) = 21.26, p < .001$). This interaction was due to the fact that there was a preference for orderings with adjectives that are needed (and sufficient) for reference resolution in first position (i.e. an effect of RELEVANCE) and this preference was more pronounced in *sharp* ($\chi^2(1) = 385.91, p < .001$) as compared to *blurred* SIZE DISTRIBUTIONs ($\chi^2(1) = 222.49, p < .001$). Since we had specific expectations concerning the effect of SIZE DISTRIBUTION on the preference for orderings with subjective adjectives in first position, we split the data according to the factor COMBINATION and performed separate analyses on *dimension and X* and *color and form* combinations. In both cases, the interaction between RELEVANCE and SIZE DISTRIBUTION turned out to be significant but for different reasons: In combinations of *dimension and X*, the preference for subjective-first orderings in dimension-relevant contexts was increased in *sharp* as compared to *blurred* DISTRIBUTIONs ($\beta = 0.58, \chi^2(2) = 19.50, p < .001$). In combinations of *color and form* adjectives, *sharp* in comparison to *blurred* distributions led, by contrast, to an increased preference for form-first orderings (the 2nd property in the COMBINATION *color and form*) in form-relevant contexts ($\beta = -0.71, \chi^2(2) = 8.29, p = 0.016$). The former of these two interactions was directly relevant to our hypotheses whereas the latter was completely unexpected and we do not have an explanation for it.

4.3 Discussion

We replicated both the SUBJECTIVITY and DIS-

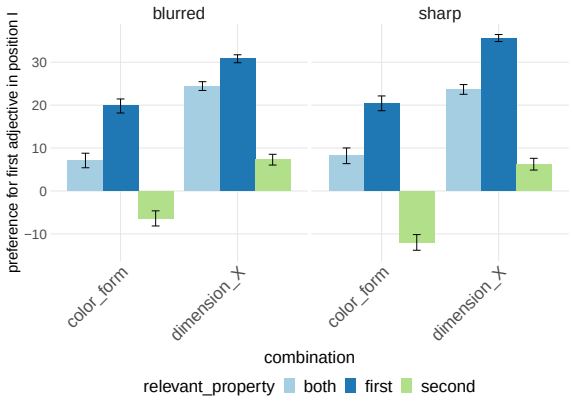


Figure 2: The mean slider positions from the current experiment: The slider had an initial value of 0 and potential values ranged between +50 and -50. A positive value indicates a preference for the first adjective in a COMBINATION (i.e. the color adjective in the combination *color and form* or the dimension adjective in the combination *dimension and x*, where x stands for either color or form) at the first position in the linear sequence. For combinations involving dimension adjectives (labeled *dimension_x*), a positive value indicates the conventional subjective-first order and a negative value shows the opposite.

CRIMINATORY STRENGTH effects in our study, which suggests that more than one source can contribute to adjective ordering preferences, especially in visual contexts. We manipulated the communicative efficiency of subjective adjectives by varying discriminatory strength of the size property and varying size distributions of contrast objects in visual contexts. Contrary to the predictions we derived from Scontras et al. (2019), our present results indicate that the robust preference for subjective-first orderings cannot be easily explained by communicative efficiency alone (cf. section 3).

5 Previous modeling approaches

Below, we propose a novel incremental model of interpretation in the RSA framework (Frank and Goodman, 2012; Scontras et al., 2018) in order to account for qualitative aspects of our experimental findings. In doing so, we build on previous models, but also highlight differences between the current and previous approaches.

In order to explain subjectivity-based ordering preferences, computational models of communication were used in recent research. The model we propose in the following section builds on some of these previous proposals (in particular, Simonic, 2018, Scontras et al., 2019 and Franke et al., 2019) that are closely related in spirit to referential communication in the RSA framework (but see also Hahn et al., 2018, for a slightly different approach). The general agreement among these approaches is that less subjective content is more effective in conveying intended meanings because it is more likely to be interpreted in the same way by listeners and speakers. Among the mentioned approaches, Franke et al. (2019) is closest to the standard, vanilla RSA model and it thus serves as a reference point for us.

Furthermore, the model of Cohn-Gordon et al. (2019) is also directly relevant for the current work. In their model a literal listener constructs meanings incrementally at each word by considering all possible completions of the sentence. This type of incremental RSA model was also combined with a continuous semantics (as proposed by Degen et al., 2020) to account for the tendency of English speakers to produce more over-specified expressions with color adjectives than with size adjectives (Waldon and Degen, 2021). However, while these incremental models can address some aspects of the production of referring expressions, they do not directly address ordering preferences for multiple adjectives and, in fact, cannot account for them for reasons we explain below.

6 A fully incremental model of interpretation

Both the SUBJECTIVITY hypothesis and the DISCRIMINATORY STRENGTH hypothesis explain ordering preferences by means of incremental processes. They differ, however, in the perspective they take. The SUBJECTIVITY hypothesis takes the perspective of a listener who performs a sequentially intersective context update in order to identify an intended referent. By contrast, the DISCRIMINATORY STRENGTH hypothesis takes the perspective of an incremental speaker who maximizes information at each step in the word-by-word production of an utterance. In order to see whether these two perspectives (combined or separately) can account for the effects we observed in our preference rating experiment, we implemented a version of an incremental listener as well as an incremental speaker in a fully incremental probabilistic computational model in the RSA framework and compared qualitative modeling results to our empirical

observations. In particular, we compared the listener and speaker perspectives and asked whether one of them or both in combination can account for our qualitative results. In what follows, we focus on the experimental conditions involving dimension adjectives because all relevant effects were found in these conditions. Furthermore, we do not distinguish between color and shape adjectives as we did not find significant differences between them when they were combined with dimension adjectives.

In the vanilla RSA model (Frank and Goodman, 2012; see Scontras et al., 2018 for review), the literal listener, L_0, infers an intended referent r by combining prior expectations, $P(r)$, about what the referent will be with the literal meaning, $[\![u]\!]$, of an utterance u according to the proportionality in (2). The listener thus updates prior expectations by filtering out all potential referents that are incompatible with the literal meaning of the utterance. The speaker, S_1, on the other hand, tries to maximize communicative utility by trading off the information an utterance provides about the intended referent (measured in its surprisal $-\log(L_0(r|u))$) against its production cost, $C(u)$. This is done by choosing utterances according to the soft-max decision rule in (2-b), where α determines how rational a speaker is in choosing between utterances.

(2) a. $L_0(r|u) \propto [\![u]\!](r) \cdot P(r)$
b. $S_1(u|r) \propto \exp(\alpha \cdot (\log L_0(r|u) - C(u)))$

We extend the vanilla RSA model in a number of ways to account for our empirical observations. The main innovations are (i) a fully incremental literal listener, who performs a sequentially intersective context update that respects the hierarchical structure underlying semantic composition (i.e. it interprets German multi-adjective sequences from right to left), and (ii) a fully incremental speaker, who produces one word after the other (from left to right). In principle, these two innovations allow us to capture ordering preferences because they break the symmetry that is usually assumed in the compositional operations used to interpret multi-adjective sequences. In contrast to previous incremental approaches (Cohn-Gordon et al., 2019; Waldon and Degen, 2021), we propose a model that allows for truly incremental processing without the need to anticipate possible sentence completions.

The incremental literal listener is defined in the recursion in the first two rows in Table 1. Applied to a single-word utterance, this is just the standard literal listener from the vanilla RSA model, with the added feature of potentially context-dependent meanings. In particular, it allows for word meanings that vary with the support of the prior probability over possible states (i.e. a distribution over potential referents in our case), $P(r)$. This feature is important for two reasons.

Firstly, gradable adjectives are well-known to have context-dependent interpretations, which have been accounted for in previous computational models in various ways (e.g. Lassiter and Goodman, 2017; Qing and Franke, 2014). Here, we adopt the so-called k%-semantics in (3-a) because it has been shown in previous work (Schmidt et al., 2009; Cremers, 2022) to match speakers' judgments remarkably well and allows for a comparison with Franke et al. (2019), who used this semantics as well. Under this semantics an individual is considered tall if its height exceeds that of k% of the individuals in the comparison class C. The k% semantics was combined with a 'perceptual blur' such that perceived sizes deviated from the ground truth according to the Weber-Fechner law (implemented as in van Tiel et al., 2021). For color adjectives, we assumed the continuous semantics in (3-b) as proposed by Degen et al. (2020). According to (3-b) categorization is imperfect in the sense that blue objects may be judged as non-blue with probability ϵ and vice versa. In the following, a relatively low value of .02 was assumed for ϵ throughout.

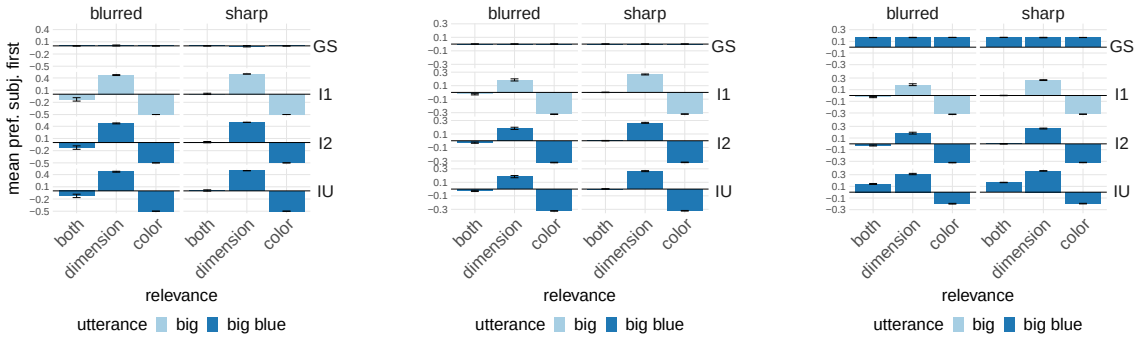
(3) a. $[\![\text{big}]\!]^C = \lambda x.size(x) > max(C) - k/100 * (max(C) - min(C))$
b. $[\![\text{blue}]\!] = \lambda x. \begin{cases} 1-\epsilon & \text{if x is blue,} \\ \epsilon & \text{if x is not blue} \end{cases}$

Secondly, the definition in Table 1 implies that the incremental listener cannot distinguish between different orders if none of the involved meanings depends on the result of the previous step in the sequential update. As a sanity check, we have verified this theoretical result by treating dimension adjectives exactly as color adjectives, using the semantics in (3-b) for them as well.

The global speaker in Table 1 functions as in the vanilla RSA model but produces utterances according to a utility function $\mathbb{U}(\overrightarrow{w}; r)$ (row 7 in Table 1) that is based on the incremental listener. This global speaker contrasts with the incremental sequence speaker, defined in rows 4 and 5 of the table, which maximizes informativity at each word. The latter is a probabilistic speaker that pro-

(1)	Incremental Listener	$L_0^{inc}(r\|w_{1,n})$	$\propto [\![w_1]\!]^{\text{supp}(L_0^{inc}(\cdot\|w_{1,n-1}))}(r) \cdot L_0^{inc}(r\|w_{1,n-1})$
(2)		$L_0^{inc}(r\|w_1)$	$\propto [\![w_1]\!]^{\text{supp}(P)}(r) \cdot P(r)$
(3)	Global Speaker	$S_1(w_{1,n}\|r)$	$\propto \mathbb{U}(w_{1,n}; r) \cdot P(w_{1,n})$
(4)	Incremental Sequence	$S_1^{inc}(w_{1,n}\|r)$	$\propto \mathbb{U}(w_{1,n}; r) \cdot P_{Lang}(w_n\|w_{1,n-1}) \cdot S_1^{inc}(w_{1,n-1}\|r)$
(5)	Speaker	$S_1^{inc}(w_1\|r)$	$\propto \mathbb{U}(w_1; r) \cdot P_{Lang}(w_1\|\varnothing)$
(6)	Incremental Utterance Speaker	$S_1^{inc\_utt}(w_{1,n}\|r)$	$\propto \exp(\alpha \cdot (\log(S_1^{inc}(w_{1,n}\|r)))) \cdot P(w_{1,n})$
(7)	Utility	$\mathbb{U}(\overrightarrow{w}; r)$	$= \exp(\beta \cdot (\log(L_0^{inc}(r\|\overrightarrow{w})) - \text{c}(\overrightarrow{w})))$

Table 1: Model definitions for the Incremental Listener (rows: 1 & 2), the Global Speaker (row: 3; GS in Fig. 3), the Incremental Sequence Speaker (rows: 4 & 5; I1 and I2 in Fig. 3), and the Incremental Utterance Speaker (row: 6; IU in Fig. 3). All speaker models depend on the utility function $\mathbb{U}$ in (7). In all the definitions, r stands for a referent; w_1, $w_{1,n}$ and $\overrightarrow{w}$ stand for the first word in a sequence, a sequence of n words and any sequence of one or more words, respectively; $\text{supp}(\cdot)$ denotes the support of a probability distribution; P denotes prior probabilities over referents and utterances; P_{Lang} assigns prior probabilities to potential next words in a sequence; and, finally, α and β are rationality parameters that govern the soft-max functions defined in rows (6) and (7), respectively. The parameter β was set to 1 in all reported simulations. In addition we used a bias (b in Fig. 3) in the prior $P(w_{1,n})$ of $S_1^{inc\_utt}$. The bias determines how much more likely the subjective-first ordering is *a priori*.



(a) $\alpha = 5; b = 1$ (b) $\alpha = 1; b = 1$ (c) $\alpha = 1; b = 2$

Figure 3: Simulations of preferences for the experimental stimuli (labeling of conditions as in Fig. 2) with different values of α, and the bias for subjective-first orders, b. In each plot, the first row shows results for the global speaker, the second and third row represent the sequence speaker distributions for one- and two-word sequences, respectively, and the fourth row represents the incremental utterance speaker. The y-axes show probabilities shifted to $[-.5, .5]$.

duces n-word sequences by recursively sampling from a sequence speaker for length $n-1$, generating a continuation word and evaluating this, as before, using the utility function $\mathbb{U}(w_{1,n}; r)$. The next word in each step is generated by a language model, $P(w_n|w_{1,n-1})$, that is extremely simple in the present case: It produces either a dimension or color adjective as the first word and then generates the other alternative in the next step. Thus, our two candidate utterances *big blue* and *blue big* are generated with equal frequency prior to factoring in the utility function. Finally, the incremental utterance speaker chooses between alternative utterances by sampling from a prior distribution over candidate utterances (*big blue* and *blue big* in our case) and reweighing their probabilities according to the sequence speaker. In the utterance prior, we used a bias parameter, b, to encode an *a priori* preference for the subjective-first ordering.

6.1 Results and discussion

The model was implemented and simulated using the probabilistic programming language WebPPL (Goodman and Stuhlmüller, 2014). We applied the model to all our stimuli from the conditions that involved dimension adjectives and tested various parameter settings. We report those that best represent the general picture that emerged. We did not find significant deviation from this general pattern for any of the parameter sets we tried. Posterior distributions were inferred using MCMC simulation with 30000 samples (burn-in: 5000, lag: 3) for the incremental listener and sequence speaker and 15000 samples (burn-in: 3000, lag: 3) for the two

utterance speakers. All simulations had an ϵ of .02 for the semantics of color adjectives, a k of 50 for the dimension adjective and a Weber fraction of .5 for the perceptual blur.

In a first simulation, we chose a relatively large value for the rationality parameter, namely $\alpha = 5$, and assumed no bias for the subjective-first order in the utterance speakers (i.e. $b = 1$). The results of this simulation are shown in Figure 3a. We did not find any deviation from uniform preferences in the global speaker (whose preferences are determined by the incremental literal listener alone). In contrast, the other three components (i.e. the sequence speaker for one- and two-word sequences and the incremental utterance speaker) revealed effects of SIZE DISTRIBUTION and also showed the characteristic effects of DISCRIMINATORY STRENGTH. [2] The effect of SIZE DISTRIBUTION was more pronounced in the conditions in which both properties were relevant than in conditions in which only one was relevant. This was because the preferences were at ceiling in the latter four conditions, revealing strong effects of discriminatory strength. Nevertheless, there was still a small effect of SIZE DISTRIBUTION in the conditions in which the dimension adjective was relevant, matching another aspect of our empirical observations.

To attenuate preferences in the conditions in which only one adjective was relevant for reference resolutions, we ran the same simulation with lower α. The results are shown in Fig. 3b. As before, effects are limited to the incremental speaker components of our model and there are again effects of both SIZE DISTRIBUTION and DISCRIMINATORY STRENGTH. As expected, extreme preferences are attenuated compared to the first simulation. This led to a preference pattern in which the effect of SIZE DISTRIBUTION is almost completely restricted to the dimension relevant conditions. Besides this effect, there are still relatively large effects of DISCRIMINATORY STRENGTH. Both of these aspects match our empirical observations. The absolute preferences, on the other hand, do not. This can, e.g., be seen by the negative values in the color-relevant and balanced preferences in the both-relevant conditions.

Absolute preferences were adjusted in a third simulation using a bias of $2 : 1$ ($b = 2$) for the subjective-first order. The resulting preferences are shown in Fig 3c. They matched our empirical observations better but still not perfectly. One notable deviation from our empirical observations consists in preferences for the subjective-last order in the color relevant conditions.

While it would be possible to shrink this deviance further using yet different parameter values, we think that this is beyond the scope of the present qualitative analysis. What our result provide, though, is initial indication concerning the region of the parameter space that may be worth examining further in a quantitative analysis. One first step towards such an analysis would be to specify a linking function between the production preferences of the model and the slider values we observed in the experiment. Their relationship may well be non-linear and could thus lead to compressed slider values in some regions.

One surprising result is that we did not find any effects whatsoever in the incremental listener component of the model. We investigated this issue further in two directions. Firstly, we used a different semantics for the dimension adjectives when modeling our experimental stimuli. This semantics was based on the identification of large and small objects based on the optimal breaks algorithm of Jenks (1967) akin to the cluster-based semantics in Schmidt et al. (2009). Secondly, we generated up to 350 random stimuli by sampling sizes from a Gaussian and colors from a Binomial distribution and modeled these stimuli using various sets of parameters (e.g. larger values of α and different values for k). We did, however, not find pronounced preferences in any of these attempts. We see two potential reason for this discrepancy between previous models (Simonic, 2018; Scontras et al., 2019; Franke et al., 2019) and the current results: It could be due to limited sample size in the present simulations or to the fact that previous models implemented different assumptions, (e.g. applying a threshold-based semantics also to color adjectives, as in Franke et al., 2019).

7 General discussion and outlook

We showed that a qualitative account of our data can be given by means of an incremental speaker that maximizes informativity at each word in combination with a general preference for subjective-first sequences. This does not imply that the general preference for subjective-first sequences is not driven by pressures towards efficient communica-

[2] We refrain from reporting statistical analyses because we did not perform a quantitative analysis and existing qualitative effects can be boosted by increasing rationality parameters.

tion in sequential context updates, as was proposed by previous studies. However, as noted in section 3, an explanation along these lines has to acknowledge the type of adaptation we observed in the current preference rating experiments. In particular, participants used subjective expressions earlier in the linear sequence if they were more informative about the intended referent. Based on the current empirical and modeling results, we would like to suggest an alternative explanation of how preferences for subjective-first sequences may emerge, at least for dimension adjectives. Such adjectives are commonly thought of as being used to communicate properties that deviate from the norm. This implies that, when they are used, they tend to have high discriminatory strength and may therefore be produced early in the linear sequence.

What we did not observe in our incremental model are truly incremental effects, i.e. shifts in preferences between one word and the next. Instead, preferences were due to an utterance-level prior in combination with a tendency to start the sequence with an informative word. The reason that incremental effects did not emerge in the current setting was that there were no strong ordering preferences on the listener side that could have modulated any initial biases. Other types of incremental effects may emerge if there are different numbers of continuations depending on how an utterance was started. Such effects were discussed, e.g., by Cohn-Gordon et al. (2019) and they can be reproduced in the current model.

Previous incremental RSA models (Cohn-Gordon et al., 2019; Waldon and Degen, 2021) were based a non-incremental semantics and evaluated all possible sentence completions of a given sentence beginning at each step. This is a natural approach because compositional semantic models often only provide interpretations for complete sentences. In contrast, our listener model evaluates an utterance word-by-word from right to left in line with the assumed sequential context update of multi-adjective strings. The more general idea behind our model is to use a genuinely incremental semantics (as proposed, e.g., by Bott and Sternefeld, 2017) that implements the local evaluation of yet incomplete sentences in a systematic fashion while ensuring that the interpretation of the complete utterance will conform to its standard compositional interpretation. We view our model as an instantiation of this general approach.

An interesting question is how much of our present considerations can be extended to non-definite noun phrases, where ordering preferences seem to persist but the current informativity-based notions do not apply directly because they are tailored to referential communication and the identification of intended referents. [3] Firstly, we see no reason to rule out the possibility that the bias we assumed in order to explain the general (context-independent) preference for subjective-first orders can be extended to non-referential usages right away. Secondly, we think that considerations based on (context-dependent) informativity might also generalize to non-definite noun-phrases. From the perspective of Generalized Quantifier Theory (Barwise and Cooper, 1981), for example, a modified noun in a quantified noun phrase provides the restriction of the quantifier and the meaning of a quantified sentence, like e.g. *many of the big white bears are moving south* is determined by two specific cardinalities: that of the set of elements that are both in the restriction and in the so-called nuclear scope of the quantifier (e.g. the big white bears moving south) and that of the set of elements that are in the restriction but not in the scope (e.g. the big white bears not moving south). Obviously, the relevant sets have to be identified first in order to determine these cardinalities. Informativity-based notions may, in principle, affect the amount of errors that are expected during this process, both from the perspective of a listener performing sequentially intersective updates as well as the perspective of a speaker aiming to provide the most discriminatory information first (see van Tiel et al., 2021, for an RSA model of quantifier interpretation).

Similarly, one might wonder how the present results generalize beyond nominal modification to the modification of verb phrases or even entire propositions (see, e.g., Specht and Stolterfoht, 2023, for an experimental investigation). While some of the present considerations might generalize to such cases, they also pose significant challenges to our present approach. In particular, such modification often involves properties that are fairly abstract or intensional in nature and are, therefore, difficult to control by means of contextual manipulations. Whether the present approach can be extended to cover such cases as well thus remains to be seen.

[3] We would like to thank an anonymous reviewer for raising this question.

Acknowledgements

We would like to thank Britta Stolterfoht, Michael Franke and three anonymous reviewers for helpful discussion and comments. FS received funding from the Baden-Württemberg Ministry of Science (MWK-BW) and the Federal Ministry of Education and Research (BMBF) as part of the Excellence Strategy of the German Federal and State Governments.

References

John Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Oliver Bott and Wolfgang Sternefeld. 2017. An event semantics with continuations for incremental interpretation. *Journal of Semantics*, 34(2):201–236.

Guglielmo Cinque. 1993. On the evidence for partial N-movement in the Romance DP. *Working Papers in Linguistics, 3.2, 1993, pp. 21-40*.

Reuben Cohn-Gordon, Noah Goodman, and Christopher Potts. 2019. An incremental iterated response model of pragmatics. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 81–90.

Alexandre Cremers. 2022. Interpreting gradable adjectives: rational reasoning or simple heuristics? In *Empirical Issues in Syntax and Semantics 14*, pages 31–61, Paris.

Judith Degen, Robert D. Hawkins, Caroline Graf, Elisa Kreiss, and Noah D. Goodman. 2020. When redundancy is useful: A Bayesian approach to "overinformative" referring expressions. *Psychological Review*, 127(4):591–621.

Robert M.W. Dixon. 1982. *Where have all the adjectives gone?: and other essays in semantics and syntax (Vol. 107)*. Janua Linguarum. Series Maior. Walter de Gruyter, Berlin, New York.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Michael Franke, Gregory Scontras, and Mihael Simonic. 2019. Subjectivity-based adjective ordering maximizes communicative success. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*, pages 344–350.

Kumiko Fukumura. 2018. Ordering adjectives in referential communication. *Journal of Memory and Language*, 101:37–50.

Noah D. Goodman and Andreas Stuhlmüller. 2014. The Design and Implementation of Probabilistic Programming Languages. http://dippl.org. Accessed: 2023-2-22.

Michael Hahn, Judith Degen, Noah D Goodman, Dan Jurafsky, and Richard Futrell. 2018. An information-theoretic explanation of adjective ordering preferences. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*.

G. F. Jenks. 1967. The data model concept in statistical mapping. *International Yearbook of Cartography*, 7:186–190.

Daniel Lassiter and Noah D. Goodman. 2017. Adjectival vagueness in a Bayesian model of interpretation. *Synthese*, 194:3801–3836.

James E. Martin. 1969. Semantic determinants of preferred adjective order. *Journal of Verbal Learning and Verbal Behavior*, 8(6):697–704.

Ciyang Qing and Michael Franke. 2014. Gradable adjectives, vagueness, and optimal language use: A speaker-oriented model. In *Proceedings of the 24th Semantics and Linguistic Theory Conference*, volume 24, pages 23–41. Linguistic Society of America.

Lauren A. Schmidt, Noah D. Goodman, David Barner, and Joshua B. Tenenbaum. 2009. How tall is tall? compositionality, statistics, and gradable adjectives. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, pages 2759–2764.

Gregory Scontras. 2023. Adjective ordering across languages. *Annual Review of Linguistics*, 9(1):357–376.

Gregory Scontras, Galia Bar-Sever, Zeinab Kachakeche, Cesar Manuel Rosales Jr, and Suttera Samonte. 2020a. Incremental semantic restriction and subjectivity-based adjective ordering. In *Proceedings of Sinn und Bedeutung*, volume 24, pages 253–270.

Gregory Scontras, Judith Degen, and Noah D. Goodman. 2017. Subjectivity predicts adjective ordering preferences. *Open Mind*, 1(1):53–66.

Gregory Scontras, Judith Degen, and Noah D. Goodman. 2019. On the grammatical source of adjective ordering preferences. *Semantics and Pragmatics*, 12:7.

Gregory Scontras, Z. Kachakeche, A. Nguyen, C. Rosales, S. Samonte, E. Shetreet, Y. Shi, Elli N. Tourtouri, and N. Trainin. 2020b. Cross-linguistic evidence for subjectivity-based adjective ordering preferences. Talk presented at the workshop on Theoretical and Experimental Approaches to Modification (TExMod2020), held at the University of Tübingen.

Gregory Scontras, Michael H. Tessler, and Michael Franke. 2018. Probabilistic language understanding: An introduction to the Rational Speech Act framework. https://www.problang.org. Accessed: 2023-2-22.

Mihael Simonic. 2018. Functional explanation of adjective ordering preferences using probabilistic programming. Master's thesis, University of Tübingen.

Larissa Specht and Britta Stolterfoht. 2023. Processing word order variations with frame and sentence adjuncts in German: Syntactic and information-structural constraints. *Glossa: a journal of general linguistics*, 8(1).

Richard Sproat and Chilin Shih. 1991. The cross-linguistic distribution of adjective ordering restrictions. In Carol Georgopoulos and Roberta Ishihara, editors, *Interdisciplinary Approaches to Language: Essays in Honor of S.-Y. Kuroda*, pages 565–593. Springer Netherlands, Dordrecht.

Bob van Tiel, Michael Franke, and Uli Sauerland. 2021. Probabilistic pragmatics explains gradience and focality in natural language quantification. *Proceedings of the National Academy of Sciences*, 118(9):e2005453118.

Brandon Waldon and Judith Degen. 2021. Modeling cross-linguistic production of referring expressions. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 206–215, Online. Association for Computational Linguistics.

Hening Wang. 2022. Subjektivität vs. diskriminatorische Stärke: Eine experimentelle Untersuchung zur Adjektivreihenfolgenpräferenz im visuellen Kontext. Master's thesis, University of Tübingen.

Benjamin Lee Whorf. 1945. Grammatical categories. *Language*, 21(1):1–11.

A Glosses for example item

(4) The question below visual contexts as part of the cover story in the current experiment (see. Fig. 1)

a. Wie fragen Sie?
how ask you
'How do you ask?'

(5) ...and questions on both sides of the slider for rating

a. Brauchst du den großen blauen Aufkleber?
need you the big blue sticker
' Do you need the big blue sticker?'

b. Brauchst du den blauen großen Aufkleber?
need you the blue big sticker
' Do you need the blue big sticker?'

Language Models Can Learn Exceptions to Syntactic Rules

Cara Su-Yi Leong **Tal Linzen**
New York University
{caraleong,linzen}@nyu.edu

Abstract

Artificial neural networks can generalize productively to novel contexts. Can they also learn exceptions to those productive rules? We explore this question using the case of restrictions on English passivization (e.g., the fact that "The vacation lasted five days" is grammatical, but "*Five days was lasted by the vacation" is not). We collect human acceptability judgments for passive sentences with a range of verbs, and show that the probability distribution defined by GPT-2, a language model, matches the human judgments with high correlation. We also show that the relative acceptability of a verb in the active vs. passive voice is positively correlated with the relative frequency of its occurrence in those voices. These results provide preliminary support for the entrenchment hypothesis, according to which learners track and uses the distributional properties of their input to learn negative exceptions to rules. At the same time, this hypothesis fails to explain the magnitude of unpassivizability demonstrated by certain individual verbs, suggesting that other cues to exceptionality are available in the linguistic input.

1 Introduction

Many studies have demonstrated language models' ability to extend a generalization from a small set of examples to novel lexical items, structures, and contexts, even if the models do not always do so in a human-like way (Hupkes et al., 2020; Kim and Linzen, 2020; Lake and Baroni, 2018; McCoy et al., 2018). These studies show that models can substitute novel lexical items into rules where those items were previously unseen. At the same time, language models can sometimes *over-generalize*, for instance by producing a literal, compositional translation of idiomatic expressions like *kick the bucket* when humans would not (Dankers et al., 2022). A full evaluation of language models' generalization abilities should thus not only measure whether models can generalize when humans do, but also whether models are able to *constrain* their generalizations when humans do.

We address this question by building on a line of work that probes whether human-like acceptability judgments for argument structure alternations can be predicted from the probability distribution that a from language model defines over sentences. This studies have shown, for example, that the GPT-2 language model (Radford et al., 2019) can match human judgments about whether the dative alternation applies to a verb (Hawkins et al., 2020), and that information about which syntactic frames a verb can appear in (e.g. whether a verb participates in the SPRAY/LOAD alternation) can be recovered from the verb's contextualized representations and from sentence embeddings (Kann et al., 2019).

In this work, we evaluate models' ability to identify exceptions using the case study of the English passive.[1] The passive voice is highly productive in English; most strikingly, young children exposed to novel verbs in the active voice are able to understand and produce passive constructions using those verbs (Pinker et al., 1987; Brooks and Tomasello, 1999). This suggests that English speakers do not in general conclude that verbs that they have never encountered in the passive voice are unacceptable in that voice. Yet there are limits to the productivity of the English passive; examples such as (1) have been reported to be unacceptable in the passive voice:

(1) a. The vacation lasted five days.
b. *Five days was lasted by the vacation.

Sentences like (1b) are unlikely to occur productively in natural speech—just like passives of infrequent verbs. Yet even though they do not receive

[1]Data and code are available at https://github.com/craaaa/exceptions.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 133-144.
Amherst, Massachusetts, June 15-17, 2023

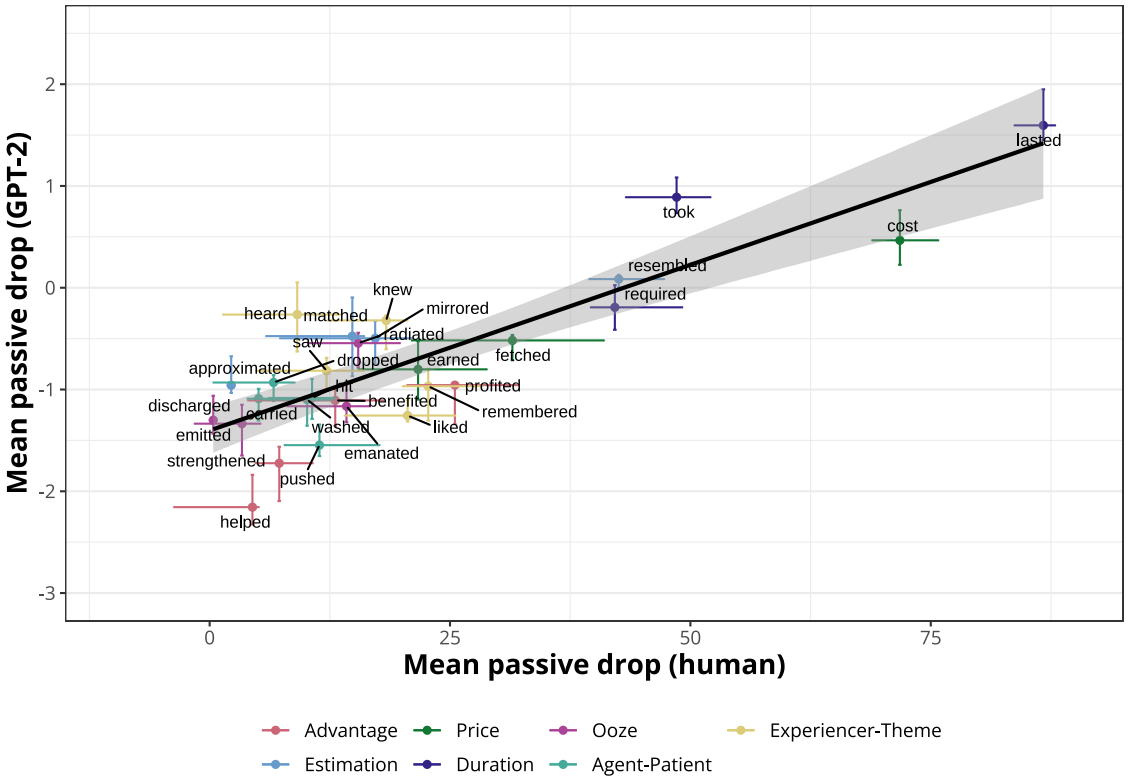


Figure 1: *Passive drop in humans vs. GPT-2* — A GPT-2 model trained on 100M words approximately predicts variable amounts of passive drop equivalent to human judgments. Horizontal and vertical error bars indicate bootstrapped 95% confidence intervals.

explicit evidence that these sentences are unacceptable, rather than simply rare, English speakers nonetheless learn that they constitute exceptions, and do not judge (1b) to be acceptable.

How do humans acquire such exceptions? The **entrenchment hypothesis** suggests that speakers track and use the distributional properties of their input as indirect negative evidence for the existence of an exception (Braine and Brooks, 1995; Regier and Gahl, 2004; Theakston, 2004). For instance, if an English learner never encounters the verb *last* in the passive voice, despite having seen *last* used productively in the active voice, they may conclude that *last* cannot occur in the the passive voice. Are language models—which do not have access to human feedback or syntactic supervision, and are trained solely to perform next-word prediction—attentive to the same information that humans are when determining the extent to which syntactic rules can generalize?

In this paper, we tackle these questions by comparing human acceptability judgments on sentences containing verbs that are exceptional in the passive voice, on the one hand, to the probability distribution defined by a GPT-2-like model trained on a 100-million word English corpus. We find that the language model matches human acceptability judgments on active and passive sentences to a large degree (Figure 1), suggesting that language models can constrain their syntactic generalizations in a human-like way. Using our model's training corpus, we further show that there is a weak but positive correlation between the relative frequency of actives and passives in the input and their relative acceptability. Together, these empirical results suggest that the linguistic input contains useful information from which exceptions to syntactic generalizations can be learned.

2 Restrictions on passivization

Although the English verbal passive is highly productive, not all verbs can occur in the passive. For

Verb class	Active sentence	Passive sentence
Advantage	Your investment ___ the community.	The community was ___ by your investment.
Price	Your book ___ thirty dollars.	Thirty dollars was ___ by your book.
Ooze	That machine ___ a sound.	A sound was ___ by that machine.
Duration	The journey ___ three days.	Three days was ___ by the journey.
Estimation	Your drawing ___ her likeness.	Her likeness was ___ by your drawing.

Table 1: *Example sentence frames* — Each verb in the verb class was substituted into frames specific to the class.

instance, intransitive and middle verbs resist passivization in general (Perlmutter, 1978; Zaenen, 1993). In this paper, we focus on passives of transitive verbs that occur with by-phrases. These long passives are clauses of the form given in (2), which in most cases have an uncontroversially acceptable passive form:

(2) a. The ball was hit by the boy.

A small list of lexical exceptions have been described for which the passive voice is deemed ungrammatical (Levin, 1993; Postal, 2004). Some of these exceptions can be classed together based on the semantics of the verb or types of arguments the verb takes. For instance, verbs that take measure phrases as their object reportedly do not occur in the passive:

(3) a. That house costs fifty thousand dollars.

b. *Fifty thousand dollars is/are cost by that house.

(Hale and Keyser, 1997, 17-8)

Even within a particular verb class, passivizability may also be an idiosyncratic characteristic of individual lexical items (Zwicky, 1987): verbs which can be substituted for each other in any other syntactic context may differ in their ability to passivize. Thus, for instance, although in the active voice *matched*, *mirrored*, *approximated* and *resembled* can occur in the same environment, (4a) is grammatical, while (4b) is not.

(4) a. Kim is matched/mirrored/approximated by the model in nearly every detail.

b. *Kim is resembled by the model in nearly every detail. (Zwicky, 1987)

We may thus expect differences in passivizability not only between verbs with different semantics and argument frames, but also among verbs with very similar meaning.

3 Human Acceptability Judgments

In order to test whether language models follow a human-like generalization patterns, we need to first characterize the human judgment pattern, which will serve as the target of modeling. In this section, we report on an acceptability judgment study whose goal was to verify the judgments from the syntax literature and measure any gradient differences in the degree to which different verbs can be passivized.

3.1 Materials

We identified five verb classes containing verbs that have been reported to be unpassivizable (Levin, 1993; Postal, 2004; Zwicky, 1987):

- Advantage verbs: *benefit*, *help*, *profit*, *strengthen*
- Price verbs: *cost*, *earn*, *fetch*
- Ooze verbs: *discharge*, *emanate*, *emit*, *radiate*
- Duration verbs: *last*, *require*, *take*
- Estimation verbs: *approximate*, *match*, *mirror*, *resemble*

Each of these class includes verbs with similar semantics that can be substituted into the same position in a sentence in the active voice. While some of these verbs can be used in other senses, we tested the specific sense that was reported in the literature by controlling the sentence frames used. Five past-tense sentence frames were constructed for each verb class (Table 1).

Each of the verbs in the class was substituted into the sentence frame, resulting in 90 total test sentence pairs. Example (5) demonstrates a sentence pair generated from the sentence frame in Table 1 using the verb *matched*:

(5) a. Your friend matched my brother.

b. My brother was matched by your friend.

As control verbs, we also selected five agent-patient and five experiencer-theme verbs; we expect these verbs to be passivizable:

- Agent-Patient: *hit*, *push*, *wash*, *drop*, *carry*
- Experiencer-Theme: *see*, *hear*, *know*, *like*, *remember*

Because of the varied semantics of the verbs in these groups, unique sentence pairs were created for each verb, yielding 50 control sentence pairs. An example of a sentence pair for the verb *push* is given in (6):

(6) a. A boy pushed the cup.

b. The cup was pushed by a boy.

Each participant only saw either the active or the passive of a sentence pair. The 140 sentence pairs (90 test + 50 control) were divided into two buckets of 70 sentence pairs each such that each bucket contained two or three sentence frames per verb. Each bucket was then further divided into groups of 70 sentences such that the active and passive forms of a sentence pair were in different groups. Each group of sentences contained one quarter of the test and control stimuli (70 sentences).

Presentation order was counterbalanced by making four ordered lists for each group. Each group was organized into two lists such that an item that appeared in the first half of of one list appeared in the second half of the other list. The order of items was pseudorandomized within those lists to ensure that not more than two active or passive sentences and no two sentences within the same verb class were seen in succession. These lists were then reversed, so that a total of four ordered sentence lists were made per sentence group.

Additionally, every experimental trial alternated with a filler sentence. Filler sentences were also used as attention checks. We used 24 grammatical and 46 ungrammatical filler sentences: since the passives of control sentences were expected to be acceptable, the greater number of ungrammatical fillers was intended to balance the experimental stimuli. The full set of materials is available in Appendix A.

3.2 Participants

We recruited 84 participants who had IP addresses located in the US and self-reported as native English speakers via the crowdsourcing platform Prolific. Each participant rated 140 sentences (70 test + 70 filler) and was paid US$3.50. The experiment took approximately 12 minutes to complete.

Participants were asked to rate how acceptable each sentence sounded based on their gut reaction. They were told that there were no right or wrong answers. Participants rated sentences by moving a slider from “Completely unacceptable" to “Completely acceptable", which corresponded to an integer score (invisible to them) between 0 and 100. They were not able to rate a sentence with a score of 50. Two practice sentences (one ungrammatical, one grammatical) were used to familiarize participants with the paradigm.

Participants were excluded from the results if they answered more than 15 filler questions unexpectedly, either by giving ungrammatical sentences scores above 50 or giving grammatical sentences scores below 50. We excluded 10 participants from analysis for this reason.

3.3 Results

We calculate the **passive drop** of an item as the difference in mean acceptability ratings between its active and passive version. The results are reported in Figure 2; a steeper downward gradient corresponds with a larger passive drop. Since corresponding active and passive sentences contain the same lexical items except for the auxiliary *was* and *by*, which are common across all sentences, directly comparing active and passive sentences isolates the effect of the passivization from lexical effects that might increase the acceptability of sentences with more common verbs like *helped* over low-frequency verbs like *profited*.

Across all verb classes, there was a significant difference between scores given to active and passive sentences. This difference may be accounted for by pragmatic factors: although the passive construction is more pragmatically marked than the active (Comrie, 1988), each sentence in the acceptability judgment task was presented to participants without establishing a relevant context. This setting might have caused participants to rate passive sentences as worse than their active counterparts.

Although the passive drop was positive for all verbs, its **magnitude** differed across verb classes. The duration class showed the largest mean passive drop (59.4 points), and the ooze class showed the lowest mean passive drop (8.0 points) among the test verb classes.

We fit a linear mixed-effects model to predict SENTENCE SCORE using the agent-patient verb class as the baseline. We used SENTENCE TYPE

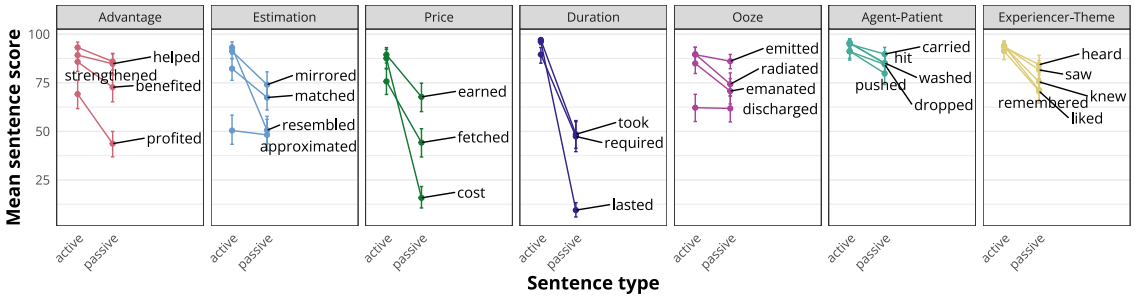


Figure 2: *Passive drop in human acceptability judgments of active and passive sentences by verb* — The steeper the downward gradient between active and passive conditions, the larger the passive drop. Error bars indicate bootstrapped 95% confidence intervals.

and VERB CLASS as well as their interaction as fixed effects and FRAME, PARTICIPANT and VERB as random intercepts. We found a significant difference between agent-patient verbs and three other verb classes: estimation verbs ($p = 5.74\text{e-}06$), price verbs ($p < 2\text{e-}16$), and duration verbs ($p < 2\text{e-}16$). On the other hand, there was no significant difference in the sentence scores obtained from agent-patient verbs and ooze verbs, advantage verbs, or experiencer-theme verbs as a class.

Within each verb class, some verbs were more passivizable than others. For example, *last* was significantly less passivizable than *took* and *required*, and *cost* was less passivizable than *fetched*. Similarly, while *resembled* had a high passive drop, the remaining verbs in the estimation class showed relatively low passive drops. These results validate the claim that some verbs may be more passivizable than others despite sharing similar paradigmatic relationships (Zwicky, 1987).

In summary, the human acceptability judgment experiment demonstrated that some verbs in the verb classes being tested are degraded in the passive voice, and that unacceptability was gradient between verbs. For a model to adequately approximate such behaviour, it must exhibit the following characteristics:

- **Exceptionality**: some verbs (e.g. duration verbs) exhibit passive drops that are significantly different from the baseline passive drop expected of the canonically passivizable agent-patient verbs.
- **Gradience**: (un)acceptability is gradient, with some verbs on average exhibiting higher passive drop than others.

4 Comparison with Language Models

With the quantitative human acceptability judgment data in hand, we now turn to evaluating language models. If distributional data is sufficient to learn the extent to which verbs are unacceptable in the passive, we expected GPT-2 to be able to match human judgments on both passivizable verbs and unpassivizable verbs. We also expect GPT-2 to be able to match the relative gradience of passive drop that humans display.

4.1 Method

We evaluated GPT-2 (Radford et al., 2019), a Transformer (Vaswani et al., 2017) language model. We tested four different pre-trained GPT-2 models, which differed in their number of parameters and number of layers, but were trained on the same data. Each model was trained on Open-AI's WebText corpus, which contains 40GB of data — approximately 8B words, assuming each word contains an average of 5 bytes/chars. Pre-trained GPT-2 models have performed well on targeted syntactic evaluations requiring knowledge of argument structure, such as differentiating between verbs that participate in the causative alternation and those that do not (Warstadt et al., 2020).

The GPT-2 models available for download are trained on a much larger corpus than is realistic for any human to be exposed to (Linzen, 2020). English-speaking children are exposed to 2–7M words per year (Gilkerson et al., 2017), or 26M–91M words by the age of 13. Rounding to the nearest order of magnitude, we trained a GPT-2 model on a 100M word subset of the OpenWebText corpus (Gokaslan and Cohen, 2019), an open-source reproduction of the Web Text corpus; this simulates more closely the amount of linguistic input

a human may receive (though not its genre). We trained five iterations of this model, which we call **GPT2-100M**, using different random seeds and report averages of the results obtained from these five models.

We adapted the targeted syntactic evaluation paradigm (Linzen et al., 2016; Lau et al., 2017; Warstadt et al., 2019) to compare the language models to humans. This paradigm involves obtaining model "judgments" for minimal pairs of sentences. For each sentence, a score is obtained by summing the log-probabilities assigned to each token in the sentence, which gives the probability the model assigns to that sentence. We conclude that a model's distribution is consistent with human judgments if it assigns a higher probability to the acceptable sentence than to the corresponding unacceptable one. Unlike some prior work, we collected numeric scores instead of binary acceptability judgments: we calculated a gradient passive drop of each sentence pair by subtracting the score of the active sentence from the score of its passive counterpart.

Since we compared active sentences to long passives, which contain by-phrases, every passive sentence contained two more words than its active counterpart. A sentence with more tokens will on balance be less probable than a sentence with fewer tokens; we thus normalized each sentence score by dividing it by the number of tokens in the sentence (Lau et al., 2017). Doing so accounts for the effect of sentence length on the sentence score, and also allows us to compare sentences where words are split into separate tokens during GPT-2's tokenization process, e.g. approximated → approx + imated.

4.2 Results

The four pre-trained models as well as the five GPT2-100M models showed positive correlations between mean human passive drop and mean model passive drop, reported in Table 2. For pre-trained GPT-2 models, we calculate mean model passive drop for each verb by averaging over the passive drop of all five sentence frames. For GPT2-100M, we calculate the average passive drop of each verb over all sentence frames across the five versions of the model (trained with different random seeds); we report these results as **GPT2-100M-avg**.

The results were qualitative similar for all models (Figure 3); in what follows, we focus on GPT2-100M-avg, whose behaviour showed the strongest correlation with human judgments. These models are also trained on the most cognitively realistic corpus.

Model	# parameters	r_s
GPT2-100M-avg	124M	**0.709**
GPT2	124M	0.659
GPT2-med	345M	0.385
GPT2-large	774M	0.549
GPT2-xl	1558M	0.559

Table 2: *GPT2 model parameters and correlation coefficients* — in all five models, a correlation was found between human passive drop and the model's passive drop, but it was stronger for smaller models, and strongest for the models trained on only 100M words.

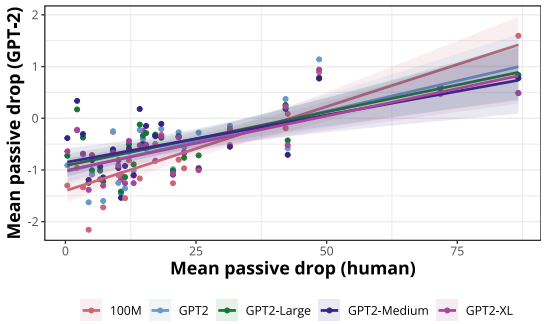


Figure 3: *Passive drop of different-sized GPT-2 models compared to human judgments* — Each point in represents a single verb. Models differed in number of parameters and/or training data, but showed qualitatively similar passive drops.

Figure 1 plots GPT2-100M-avg's passive drop against the passive drop observed in the human experiment. A strong correlation was found between the passive drop in the models' sentence scores and human passive drop (r_s = 0.709), suggesting that predictions learned from linguistic input match human **gradient** judgments on passivization relatively well.

GPT2-100M-avg also matched humans' judgments of **exceptionality** within verb classes: among verbs with similar meanings, both humans and the model identified the same verbs as being less passivizable. In verbs for which humans demonstrated low passive drop, such as *strengthened* and *discharged*, close to no passive drop was observed in the model's predictions. GPT2-100M-avg also predicted high passive drops for verbs like

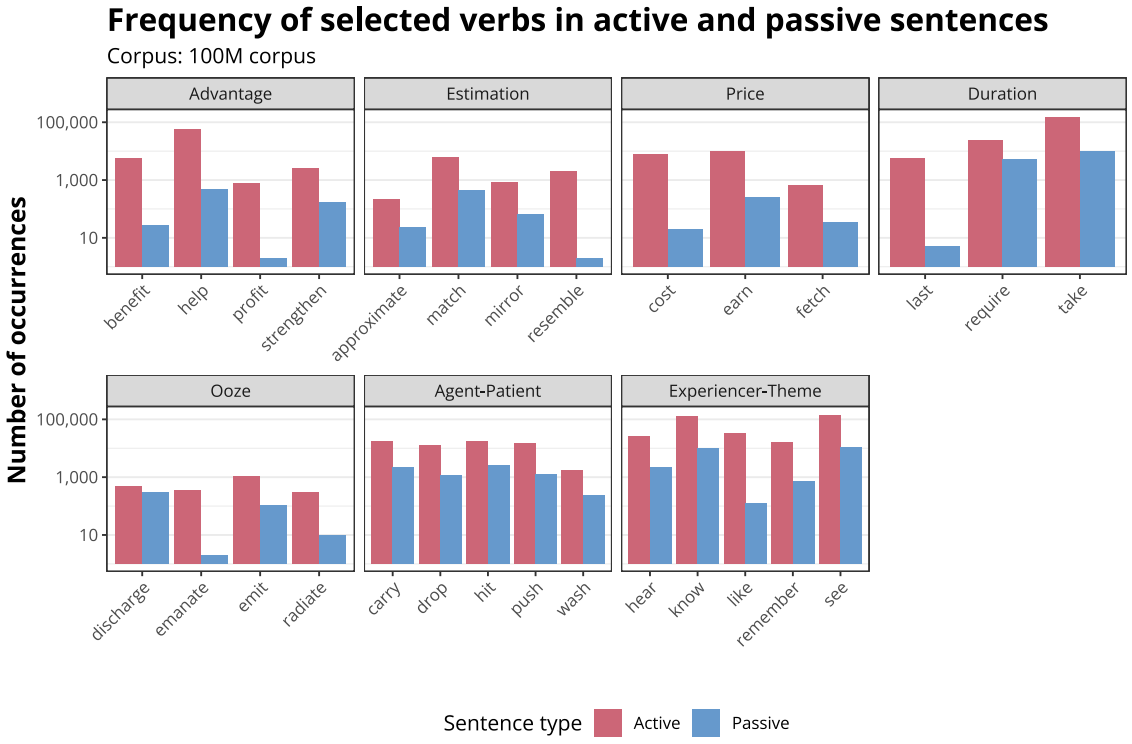


Figure 4: *Occurrence of active transitive and passive sentences using test verbs in the training corpus* — sentences whose verb had a passive dependent (`csubjpass`, `nsubjpass`, or `auxpass`) were tagged as passive, while all other instances of the verb were tagged as active.

lasted, *resembled* and *cost*, aligning with human judgments that these verbs are unique in their verb class.

5 Does Frequency Explain Passivizability Judgments?

Having established that a language model can successfully model humans' gradient passivizability judgments, we now examine the extent to which GPT2-100M's passivization judgments correlate with the distributional properties of its training data. Specifically, we explore the utility of the entrenchment hypothesis in explaining GPT2-100M's gradient judgments of passivization. Recall that this hypothesis argues that learners conclude that a verb cannot appear in a particular context if it appears in many other contexts but systematically fails to appear in the context in question.

Here, we consider a weaker version of the entrenchment hypothesis, which does not presuppose that exceptions *never* occur in the learner's input. Instead, we hypothesize that the less frequently a verb is used in the passive voice relative to the active voice, the less acceptable passive constructions using that verb will be.

5.1 Method

We conducted a corpus study on the data that GPT2-100M was trained on. We processed each document in the corpus using the spaCy Transformer-based lemmatizer, POS tokenizer and dependency parser (Honnibal et al., 2020) and extracted all sentences that contained a verbal lemma corresponding to the test and control verbs. Sentences that contained the verbs in question and had a dependency edge to a passive auxilliary (`auxpass`), a passive nominal subject (`nsubjpass`) or a passive clausal subject (`csubjpass`) were classified as passive sentences, while all other sentences containing the verb were classified as active sentences. We hand-checked a 1000 sentence subset of the training data to verify the accuracy of the tagging process. No sentences were incorrectly tagged in the manually verified subset, although the corpus did contain instances of typos such as (7) (tagged as passive):

(7) It was fun while it was lasted.

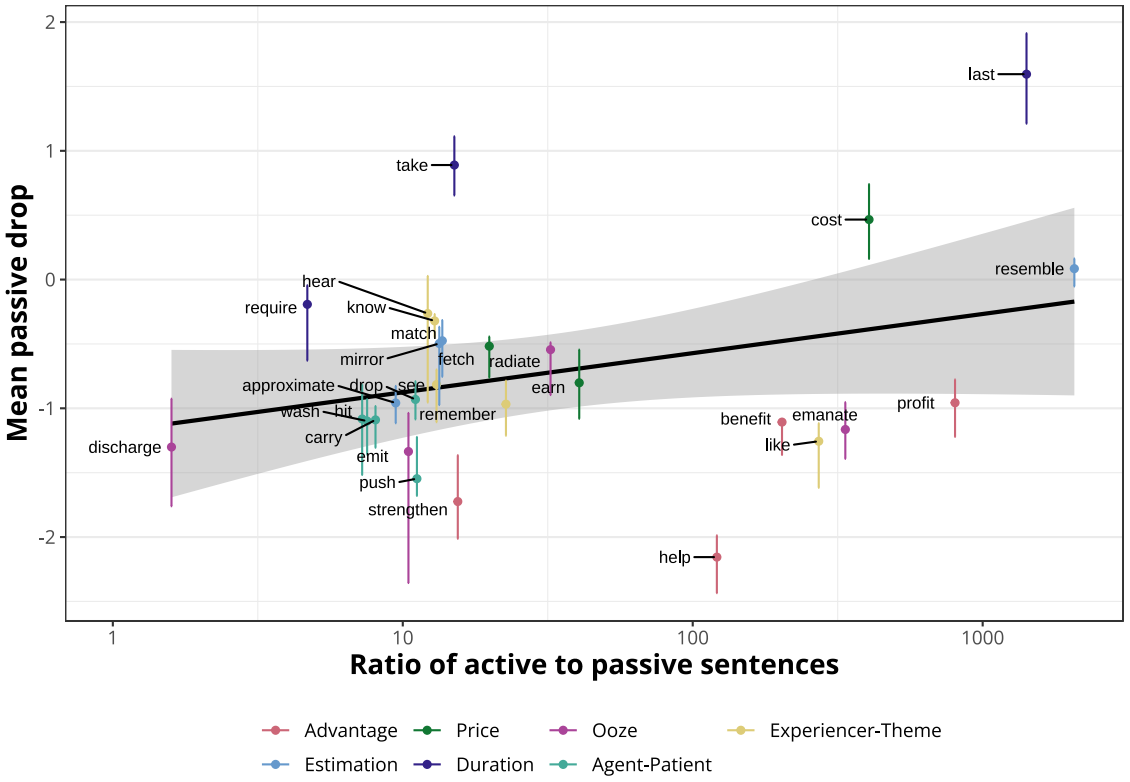


Figure 5: *GPT2-100M's passive drop against the ratio of active to passive sentences in its training corpus*. Error bars indicate bootstrapped 95% confidence intervals across sentence frames.

5.2 Results

Figure 4 shows the number of active and passive sentences in GPT2-100M's training corpus.

Not all verbs appear in the same ratios in the active and passive voice. Agent-patient verbs consistently appeared in approximately 10 times as many active sentences as passive sentences, matching estimates from previous corpus studies (Roland et al., 2007). On the other hand, test verbs appeared in varying amounts in the active and passive. For instance, *last* appeared 5666 times in the active and four times in the passive in the 100M word corpus, while *cost* appeared 7706 times in the active and 19 times in the passive. This result suggests that the test verbs differ from canonically passivizable control verbs in their distribution.

Figure 5 graphs the correlation between the ratio of active to passive sentences for a given verb, on the one hand, and that verb's mean passive drop on the other hand. We find a weak but positive correlation between the two variables (r_s= 0.212).

Two key outliers that are not well accounted for by this measure of relative frequency are *last* and *cost*. In both humans and model judgments, these verbs demonstrated high passive drops; yet, they are similar in relative frequency of active and passive to verbs like *emanate*, *profit* and *resemble*, whose passive drops are lower. While frequency seems to predict some amount of unpassivizability, then, it cannot account for the full magnitude of the passive drop displayed by these particular verbs.

Furthermore, entire verb classes are systematically over- or under-predicted in Figure 5. The duration verb class on the whole has a high passive drop relative to its frequency in the corpus, while frequency over-predicts the passive drop expected for the advantage verb class. We thus conclude that while the relative frequency of active and passive voice sentences positively correlates with passive drop, other factors are likely to also be relevant on a verb-class level.

Although *take* appears to be an outlier in Figure 5, with an active to passive ratio similar to that of the agent-patient and experiencer-theme con-

trol verbs, the measure of frequency we used does not take into account the fact that *take* has multiple senses. If a different sense than the one being tested is heavily represented by passive sentences, the number of passives counted may be overestimated. For example, although we only test the duration sense of *take*, as given in (8a), the sense used in (8b) may be more prevalent in the corpus:

(8) a. *Two days was taken by the meeting.

b. The photo was taken by the boy.

These differences in verb sense are not accounted for in the current corpus study; future work should make use of word sense disambiguation to conduct more targeted corpus analyses. Additionally, the issue of differentiating verb senses in polysemous verbs is one that both human and machine learners face, raising the question of the extent to which learners differentiate between verb senses that are more or less difficult to passivize.

Overall, while the relative frequency of a verb's occurrence in the active and passive does positively correlate with its unpassivizability, it does not account for crucial verb-level differences in the magnitude of passive drop demonstrated by GPT2-100M-avg.

6 Discussion

The goal of this study was to explore whether a language model can identify exceptions to a productive syntactic rule in a human-like way. We compared human acceptability judgments to sentence scores produced by a GPT-2 model trained on the amount of linguistic input that a human can plausibly be exposed to, and found that the model displayed human-like exceptionality and gradience in its judgments of passive sentences. The results of our study suggest that language models are able to refrain from over-generalizing to exceptions. Our results suggest that future empirical inroads may be made towards understanding the mechanisms and input required to overcome the projection problem (Baker, 1979), i.e. the problem of acquiring arbitrary negative exceptions, using language models as experimental subjects.

We took a first step in this direction by showing a positive correlation between the relative frequency of active and passive sentences containing a given verb and the difference between that verb's acceptability in the active and passive voice (i.e. its passive drop) in GPT-2. Although our results lend some credence to the entrenchment hypothesis, they suggest that additional factors must be recruited to explain the full magnitude of exceptionality displayed by highly unpassivizable verbs such as *last* and *cost*.

Moreover, although we demonstrated that the relative frequency of a verb's occurrence in the active and passive is correlated with its passive drop, a causal relationship between the two cannot be established from our data. A single underlying factor, such as verbal semantics, may affect both the frequency of a verb in the passive in relation to the passive *and* its acceptability in the passive construction.

Future research should test the causal impact of a verb's absolute and relative frequency in the training corpus on its predicted passivizability. Following Wei et al. (2021), we plan to create an altered training dataset where we match the frequency of active and passive sentences containing passivizable verbs like *drop* to the absolute frequency of sentences containing highly unpassivizable verbs, such as *last*. Comparing models trained on this dataset against GPT2-100M will allow us to move beyond a correlational analysis and explore whether altering the frequency of a verb in the active and passive voice in a model's training data has a causal effect on the model's predictions of that verb's passivizability.

7 Conclusion

In this paper, we explored whether a language model trained on a human-scale amount of linguistic input is able to learn lexical exceptions to a productive syntactic generalization in English. We showed that it was able to match humans' reported judgments on unpassivizable verbs like *last*, showing both the ability to identify exceptions as well as to identify the magnitude of an exception. We also demonstrated a weak correlation between the degree to which a model prefers active over passive sentences using a given verb, on the one hand, and the ratio between the frequencies with which sentences containing that verb occur in the active and passive voice, on the other hand. Together, these results suggest that distributional information plays a role in learning exceptions to syntactic rules.

Acknowledgments

We would like to thank Alec Marantz and Gary Thoms for valuable comments and suggestions related to this paper. This material is based upon work supported by the National Science Foundation (NSF) under Grant No. BCS-2114505. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

C. L. Baker. 1979. Syntactic Theory and the Projection Problem. *Linguistic Inquiry*, 10(4):533–581.

Martin D S Braine and Patricia J Brooks. 1995. Verb Argument Structure and the Problem of Avoiding an Overgeneral Grammar. In Michael Tomasello and William Edward Merriman, editors, *Beyond Names for Things: Young Children's Acquisition of Verbs*. L. Erlbaum, Hillsdale, N.J.

Patricia J. Brooks and Michael Tomasello. 1999. Young children learn to produce passives with nonce verbs. *Developmental Psychology*, 35(1):29.

Bernard Comrie. 1988. Passive and voice. In Masayoshi Shibatani, editor, *Passive and Voice*, Typological Studies in Language, pages 9–24. John Benjamins Publishing Company.

Verna Dankers, Christopher Lucas, and Ivan Titov. 2022. Can Transformer be Too Compositional? Analysing Idiom Processing in Neural Machine Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626, Dublin, Ireland. Association for Computational Linguistics.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, Oller D. Kimbrough, John H. L. Hansen, and Terrance D. Paul. 2017. Mapping the Early Language Environment Using All-Day Recordings and Automated Analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText corpus.

Ken Hale and Samuel Jay Keyser. 1997. Adjectives, other stative predicates and the roots of stativity.

Robert Hawkins, Takateru Yamakoshi, Thomas Griffiths, and Adele Goldberg. 2020. Investigating representations of verb bias in neural language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4653–4663, Online. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality Decomposed: How do Neural Networks Generalise? (Extended Abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 5065–5069, Yokohama, Japan. International Joint Conferences on Artificial Intelligence Organization.

Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R. Bowman. 2019. Verb argument structure alternations in word and sentence embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 287–297.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Brenden M. Lake and Marco Baroni. 2018. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge. *Cognitive Science*, 41(5):1202–1241.

Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago.

Tal Linzen. 2020. How Can We Accelerate Progress Towards Human-like Linguistic Generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2018. Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society, CogSci 2018*, Proceedings of the 40th Annual Meeting of the Cognitive Science Society, CogSci 2018, pages 2096–2101. The Cognitive Science Society.

David M. Perlmutter. 1978. Impersonal Passives and the Unaccusative Hypothesis. *Annual Meeting of the Berkeley Linguistics Society*, 4(0):157–190.

Steven Pinker, David S. Lebeaux, and Loren Ann Frost. 1987. Productivity and constraints in the acquisition of the passive. *Cognition*, 26(3):195–267.

Paul Martin Postal. 2004. *Skeptical Linguistic Essays*. Oxford University Press, Oxford ; New York.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI.

Terry Regier and Susanne Gahl. 2004. Learning the unlearnable: The role of missing evidence. *Cognition*, 93(2):147–155.

Douglas Roland, Frederic Dick, and Jeffrey L. Elman. 2007. Frequency of basic English grammatical structures: A corpus analysis. *Journal of Memory and Language*, 57(3):348–379.

Anna L Theakston. 2004. The role of entrenchment in children's and adults' performance on grammaticality judgment tasks. *Cognitive Development*, 19(1):15–34.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS 2017)*, volume 30. Curran Associates, Inc.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Annie Zaenen. 1993. Unaccusativity in Dutch: Integrating Syntax and Lexical Semantics. In James Pustejovsky, editor, *Semantics and the Lexicon*, Studies in Linguistics and Philosophy, pages 129–161. Springer Netherlands, Dordrecht.

Arnold M. Zwicky. 1987. Slashes in the passive. *Linguistics*, 25(4).

A Stimuli

A.1 Test sentence frames

Verb class	Sentence frame
Advantage	Your investment ___ the community.
	The exercise ___ his fitness.
	Our friendship ___ my life.
	The law ___ these workers.
	The treaty ___ both countries.
Price	Your dish ___ ninety dollars.
	The painting ___ a fortune.
	The tickets ___ a lot of money.
	Your book ___ thirty dollars.
	His actions ___ the medal.
Ooze	My friend ___ confidence.
	The lightbulb ___ some light.
	That machine ___ a sound.
	The teacher ___ wisdom.
	The trash ___ an odor.
Estimation	Your drawing ___ her likeness.
	Your friend ___ my brother.
	The character ___ the author.
	Her son ___ her father.
	The copy ___ the original.
Duration	The journey ___ three days.
	My meeting ___ two hours.
	The interview ___ some time.
	Her speech ___ seventeen minutes.
	His trek ___ a month.

A.2 Agent-patient sentences

Verb	Active sentence
hit	My brother hit your friend.
	Your sister hit the target.
	The child hit the ball.
	A boy hit my bag.
	A monkey hit the toy.
pushed	My brother pushed a child.
	The mother pushed my toy.
	A boy pushed the cup.
	A child pushed the bag.
	Your sister pushed your friend.
washed	A boy washed the cup.
	A child washed the bag.
	My sister washed a towel.
	My brother washed my plate.
	Your mother washed my toy.
dropped	My brother dropped my plate.
	The mother dropped my toy.
	A boy dropped the cup.
	A child dropped the bag.
	Your sister dropped a book.
carried	A boy carried my bag.
	Your mother carried the child.
	My brother carried your friend.
	The dog carried the toy.
	The donkey carried the load.

A.3 Experiencer-theme sentences

Verb	Active sentence
saw	My brother saw your friend.
	Your dog saw the toy.
	Your sister saw a book.
	A boy saw my bag.
	The child saw a monkey.
heard	A boy heard the sound.
	The child heard the rules.
	My brother heard your friend.
	Your dog heard the toy.
	Your sister heard a squeak.
knew	My brother knew your friend.
	Your dog knew my cat.
	Your sister knew my brother.
	A boy knew my mother.
	The mother knew the dog.
liked	A boy liked the game.
	The child liked a monkey.
	My brother liked your friend.
	Your dog liked the toy.
	Your sister liked a book.
remembered	My brother remembered your friend.
	Your dog remembered my toy.
	Your sister remembered a book.
	A boy remembered the game.
	The child remembered the rules.

An MG Parsing View into the Processing of Subject and Object Relative Clauses in Basque

Matteo Fiorini
University of Utah
matteo.fiorini@utah.edu

Jillian Chang
Great Neck South High School
jillianchang15@gmail.com

Aniello De Santo
University of Utah
aniello.desanto@utah.edu

Abstract

Stabler (2013)'s top-down parser for Minimalist grammars has been used to account for a variety of off-line processing preferences, with measures of memory load sensitive to subtle structural details. This paper expands the model's empirical coverage to ergative languages by looking at the processing asymmetries reported for Basque relative clauses. Our results show that the model predicts a subject over object preference as identified in the relevant psycholinguistic literature.

1 Introduction

A core question in research on human sentence processing is how language-specific linguistic features interact with more general processing mechanisms to give rise to the behavioral patterns recorded in production/comprehension experiments.

In this sense, differences between subject and object relative clauses (SRC and ORC, respectively) have received a lot of attention throughout the years (see Lau and Tanaka, 2021, for a review). Generally, SRCs are more common cross-linguistically (Keenan and Comrie, 1977), and they are reportedly produced and comprehended earlier and more easily than ORCs. While this subject advantage can be modulated by other properties of the sentence (e.g. case mismatches), it seems to be an overall strong pattern in both head-initial nominative/accusative languages with postnominal RCs (e.g., English or French; Mecklinger et al., 1995; Gibson, 1998; Frazier, 1987; Friedmann and Novogrodsky, 2004) and (somewhat less reliably) in head-final languages with prenominal RCs (e.g. Korean or Japanese; Kwon et al., 2010, 2013; Nakamura and Miyamoto, 2013).

Crucially, the very broad question about the interaction between language-specific properties and general cognitive processes of the human parser also leads to the more specialized question of *which* features of a language matter for different aspects of sentence processing, and how. In particular, from the perspective of highly detailed syntactic frameworks, it seems important to probe the relevance of fine-grained syntactic details in deriving behavioral patterns (Miller and Chomsky, 1963; Bresnan, 1978; Rambow and Joshi, 1997).

In this paper, we follow work recasting this question in computational terms, by specifying a transparent linking hypothesis between the syntactic structures assumed in Minimalism (Chomsky, 1995) and off-line processing difficulty. Specifically, we adopt a model integrating Stabler (2013)'s top-down parser for Minimalist grammars (Stabler, 1996, 2011) with complexity metrics measuring memory usage to derive off-line estimates of processing complexity, based on the interaction between the parser's tree-traversal strategy and the rich structure of a derivation (Kobele et al., 2013; Gerth, 2015; Graf et al., 2017; De Santo, 2020b).

RCs in general, and the asymmetries in processing between subject and object RCs in particular, have been extensively probed with this model across a variety of languages (Graf et al., 2015, 2017; De Santo, 2021a,b; Zhang, 2017). Here then, we contribute to this line of work by evaluating the model's ability to predict the contrast between SRCs and ORCs reported for Basque. Basque is of particular interest to this type of investigation as a highly inflected, ergative, SOV language with both prenominal relatives and postnominal RCs. Ergative languages have been somewhat generally overlooked in past psycholinguistic work on the comprehension and production of RCs (Carreiras et al., 2010; Juncal Gutierrez Mangado and José Ezeizabarrena, 2012; Yetano and Laka, 2019, a.o.), as well as in the recent MG parsing literature. Their morpho-syntactic properties, however, make them ideal candidates to explore how the properties of RCs interact with processing strategies, as they pose challenges for various syntactic accounts of sentence structure *and* theories of sentence process-

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 145-154.
Amherst, Massachusetts, June 15-17, 2023

ing proposed for other languages.

Due to its particular structural properties, Basque thus presents a novel, challenging test case for the computational model adopted in this paper. In showing how the model handles the SRC vs. ORC contrast reported by some of the Basque literature on RC comprehension, we not only extend the typological coverage of the model, but also highlight the relevance of computational models grounded in theoretical considerations in opening new research directions at the intersection between theoretical syntax and sentence processing.

2 Preliminaries: MG Parsing

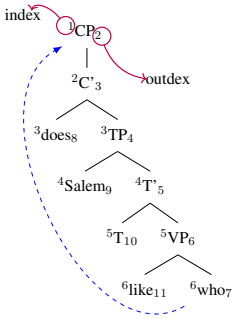
Minimalist grammars (MGs; Stabler, 1996, 2011) are a lexicalized, feature driven formalism incorporating the structurally rich analysis of Minimalist syntax (Chomsky, 1995, a.o.).

An MG is a set ot lexical items (LIs) consisting of a phonetic form and a finite, non-empty string of features. The latter are divided into two types: *Merge* features and *Move* features. LIs are assembled via the two relative feature checking operations *Merge* and *Move*. Essentially, Merge encodes subcategorization, while Move long-distance displacement dependencies. Given the scope of this paper, the technical details of the mechanism behind feature-checking are unnecessary — and in fact, in the rest of the paper we avoid displaying the feature component of the LIs altogether. What we want to highlight instead is the intuition behind the core MG data structure: *derivation trees*.

Intuitively, MG derivation trees encode the sequence of operations (Merge and Move) required to build the phrase structure tree for a specific sentence (Michaelis, 1998; Harkema, 2001; Kobele et al., 2007). Observe the tree in Figure 1b, representing a simplified derivation of the sentence *Who does Salem like?*. Here, leaf nodes are labeled by LIs, while unary and binary branching nodes represent Move and Merge operations, respectively. The main and crucial difference between this representation and a more standard phrase structure tree is that in these derivations, moving phrases remain in their base position: their landing site can be fully (deterministically) reconstructed via feature calculus. What this means though is that the final word order of a sentence is not directly reflected in the order of the leaf nodes of a derivation tree. For the sake of clarity, while movement arrows are not technically part of this representation, since

Steps	Parse Action
1	CP is conjectured
2	CP expands to C'
3	C's expands to does and TP
4	TP expands to Salem and T'
5	T' expands to T and VP
6	VP expands to like and who
7	**who** is found
8	**does** is found
9	**Salem** is found
10	**T** is found
11	**like** is found

(a)



(b)

Figure 1: Example of a string-driven top-down tree traversal for an MG derivation tree of the sentence *Who does Salem like?*.

we make away with features in the rest of the paper, we will incorporate dashed arrows to indicate movement relations.

2.1 Top-Down Parsing

Stabler (2013) takes advantage of the fact that — modulo a more complex mapping from trees to strings — MG derivation trees form a regular tree language, to propose a string-driven MG variant of a standard depth-first, top-down parser for Context-Free Grammars. Essentially, this parser hypothesises tree nodes from top to bottom and from left to right. However, since the surface order of lexical items in the derivation tree is not the phrase structure tree's surface order, simple left-to-right scanning of the leaf nodes yields the wrong order. The MG parser, while scanning the nodes, must thus also keep tracking the derivational operations which affect the linear word order and prioritizes resolving movement dependencies over the top-down strategy (i.e. the string-driven component).

Following Kobele et al. (2013), without delving too much in technical details, the parsing procedure can be outlined as follows: I) hypothesize the top of structure and add nodes downward (toward words) and left-to-right; II) if move is predicted, it triggers the search for mover $\rightarrow$ build the shortest path towards predicted mover; III) once the mover has been found, continue from the point where it was predicted. A memory stack plays a fundamental role in this: if a node is hypothesized at step i, but cannot be worked on until step j, it must be stored for $j - i$ steps in a priority queue.

The example in Figure 1a exemplifies this strategy for the tree in Figure 1b. To keep track of these operations, we follow past literature on this topic

and adopt Kobele et al. (2013)'s notation: each node in the tree is annotated with a superscript (index) and a subscript (outdex). The annotation intuitively indicates for each node in the tree I) when it is first conjectured by the parser (index) and placed in the memory stack, and II) at what point it is considered completed and flushed from memory (outdex).

Since MGs are able to closely encode the detailed structural analyses of Minimalist syntax, Stabler's MG parser has led to a rich line of work aimed at connecting syntactic assumptions to offline processing behavior, through the use of *complexity* metrics (Kobele et al., 2013; Gerth, 2015; De Santo, 2020b, a.o.).

2.2 Complexity Metrics

We employ complexity metrics that predict processing difficulty based on how memory usage is affected by the geometry of the trees built by the parser.

Building on previous work in (computational) psycholinguistics (Gibson, 1998; Rambow and Joshi, 1997, a.o.), Kobele et al. (2013) identify broad cognitive notions of memory usage like 1) *tenure*: how long a node is kept in memory and 2) *size*: the amount of information a node consumes in memory. In practical terms, the *tenure* of a node is equal to the difference between its index and its outdex. Given how derivation trees are built by the parser, given a left-to-right string to tree matching a tenure of two is the minimum expected for the right sister in a tree with binary branching — thus, tenure ≤ 2 is labelled as *trivial*. In practice, size encodes how nodes in a derivation consume memory because a phrase *m* moves across these nodes — and it can be computed in our simplified representation of derivation trees by subtracting the index of a moved element from the index of its landing site (see Graf and Marcinek, 2014; Graf et al., 2015, for a more technical discussion). For instance, referring to the annotated tree in Figure 1b, the size of *who* is 6.

These memory notions can then be used to define a large set of complexity metrics measuring the offline processing difficulty over a full derivation tree. Kobele et al. (2007) associate tenure with quantitative values by implementing complexity metrics such as: MAXT := *max*(*tenure-of(n)*), and SUMT := Σ_n *tenure-of(n)*. MAXT measures the maximum amount of time any node stays in memory during processing, while SUMT measures the overall amount of memory usage for all nodes with non-trivial tenure (i.e., > 2), capturing the total memory usage over the course of a parse. Building on these findings, Graf and Marcinek (2014) show that MAXT (only considering pronounced nodes) makes the right difficulty predictions for several phenomena, e.g., right vs. center embedding, nested vs. crossing dependencies, and the contrasts involving relative clauses at the center of our paper.

Following up on these results, Graf et al. (2015) extend the definition of these complexity measure to size. For instance, SUMSIZE can be used to measure the overall cost of maintaining long-distance filler-gap dependencies over a derivation. Let *M* be the set of all nodes of a derivation tree *t* that are the root of a subtree undergoing movement. For each $m \in M$, *i(m)* is the index of *m* and *f(m)* is the index of the highest Move node that *m*'s subtree is moved to. Then SUMSIZE is defined as $\Sigma_{m \in M}$ *i(m)* - *f(m)*.

Graf et al. (2015) also propose an idea similar to Optimality Theory's (Prince and Smolensky, 2008) constraint ranking. In their formulation, metrics of the type $\langle M_1, M_2, ..., M_n \rangle$ are ranked, and lower ranked metrics only matter if all higher ranked metrics have failed to pick out a unique winner (e.g., two constructions result in a *tie* over MAXT). While such a system would easily generate an enormous number of possible metrics, Graf et al. (2017) have argued that a small number of such metrics is in fact enough to account for a vast number of processing contrasts cross-linguistically. In particular, the ranking ⟨MAXT, SUMSIZE⟩ has been surprisingly successful in offering insights into the connection between processing load and syntactic choices (see De Santo, 2020a; Liu, 2018; Lee, 2018; De Santo and Lee, 2022, for additional support to these claims).

With a slight over-generalization of terminology, henceforth we refer to the combination of MGs as a grammar formalism, Stabler (2013)'s top-down parser, and complexity metrics estimating memory usage as the *MG Parser* or the *MG Model*, even though it is important to recognize that alternative combinations of these components are possible (Yun et al., 2015; Hunter, 2019; Hunter et al., 2019). Following Graf et al. (2017) and others, we then use this model to conduct pairwise comparisons of full derivations for constructions under analysis

(e.g., SRCs vs. ORCs) and derive estimates of processing difficulty that we can categorically match to the contrasts reported in the psycholinguistics literature.[1]

3 SRCs vs ORCs in Basque

Ergative languages, albeit representing roughly 25% of the world languages (Dixon, 1994), have received relatively little attention in computational psycholinguistics' literature. As mentioned, Basque is an ergative and head-final language allowing for prenominal RCs (de Rijk, 2007). Furthermore, Basque is a three-way pro-drop language that can omit all arguments in a sentence (i.e., the XPs marked by ergative, absolutive, and dative case). Finally, while canonically an SOV language, the word order of Basque is prone to variation (de Rijk, 2007).

Importantly, while prenominal RCs have been modelled with the MG parser (Graf et al., 2017; Zhang, 2017) the focus (somewhat in parallel with the broader psycholinguistics literature) has been on East Asian languages (Japanese, Korean, and Mandarin Chinese more specifically). The availability of prenominal RCs combined with a highly flexible word order, and ergativity, however, makes Basque an ideal candidate to expand the array of languages the MG model has been tested against.

Consider now sentences like in (1), illustrating Basque's prenominal RC constructions. This example presents an SRC (1-a) and an ORC (1-b) in a subject-modifying set-up (that is, the RC modifies a noun acting as the subject of the main clause).

(1) a. Irakasleak aipatu ditu-en ikasleak lagunak ditu orain.
teacher.PL.ABS mention.PRT has=comp student.SG.ERG friend.PL.ABS has now
'The student that mentioned the teachers has friends now.' **SRC**

b. Irakasleak aipatu ditu-en ikasleak lagunak dira orain.
teacher.PL.ABS mention.PRT has=comp student.PL.ERG friend.PL.ABS are now
'The students that the teachers mentioned are friends now.' **ORC**

Consistently with other languages with prenominal RCs, behavioral experiments on Basque RCs preferences are somewhat split (cf. Kwon et al., 2013; Yang et al., 2010; Gibson and Wu, 2013; Yetano and Laka, 2019). However, there seems to be sound evidence that, in absence of other confounds (e.g. morphological and syntactic ambiguity) Basque participants show a clear subject preference (Juncal Gutierrez Mangado and José Ezeizabarrena, 2012; Munarriz et al., 2016; Yetano and Laka, 2019).

Additionally, recent studies on Basque have also shown a strong subject preference for postnominal RCs, a construction that seems to lack some of the morpho-syntactic ambiguity present in the prenominal structure (Carreiras et al., 2010; Yetano and Laka, 2019, a.o.). However, the syntactic status of postnominal RCs in Basque is controversial and understudied to the point that we are not aware of extensive theoretical work discussing the structural details of such configuration. Since syntactic choices are fundamental to the modelling approach taken here, in the present paper we leave postnominal structures aside, and focus on evaluating whether the parser can predict a subject advantage for prenominal sentences as in (1).

4 Modelling Basque RCs

As input to the MG parser we used derivations for the sentences in (1), as shown in Figure 2 and Figure 3. We expect a preference for SRCs over ORCs (SRC > ORC), following the results in (Juncal Gutierrez Mangado and José Ezeizabarrena, 2012; Munarriz et al., 2016). As the MG parser is sensitive to fine-grained structural details, we are interested in a) capturing current Minimalist approaches to the structure of these sentences and b) explore how much particular syntactic choices involved in the derivation of RCs affect the parser's prediction. Thus, we compare derivations following two different approaches to the structure of restrictive relatives. Note also that, given the combination of an SOV base-clause plus a prenominal RC construction, these sentences show a *gap-filler*

[1] It is worth mentioning that in its full formulation, Stabler's parser exploits a search beam discarding the most unlikely predictions at each step. However, past work (starting with Kobele et al., 2013) has ignored the beam, assuming that the parser is equipped with a perfect oracle, which always makes the right choices when constructing a tree. On the one hand, such an idealization is obviously implausible from a psycholinguistics perspective. On the other hand, this choice allows the model to focus on the specific contribution of structure building operations to processing difficulty. Interestingly, even in this configuration the MG Parser has been used to gain insights into phenomena dealing with ambiguity resolution (De Santo and Lee, 2022; Lee and De Santo, 2022).

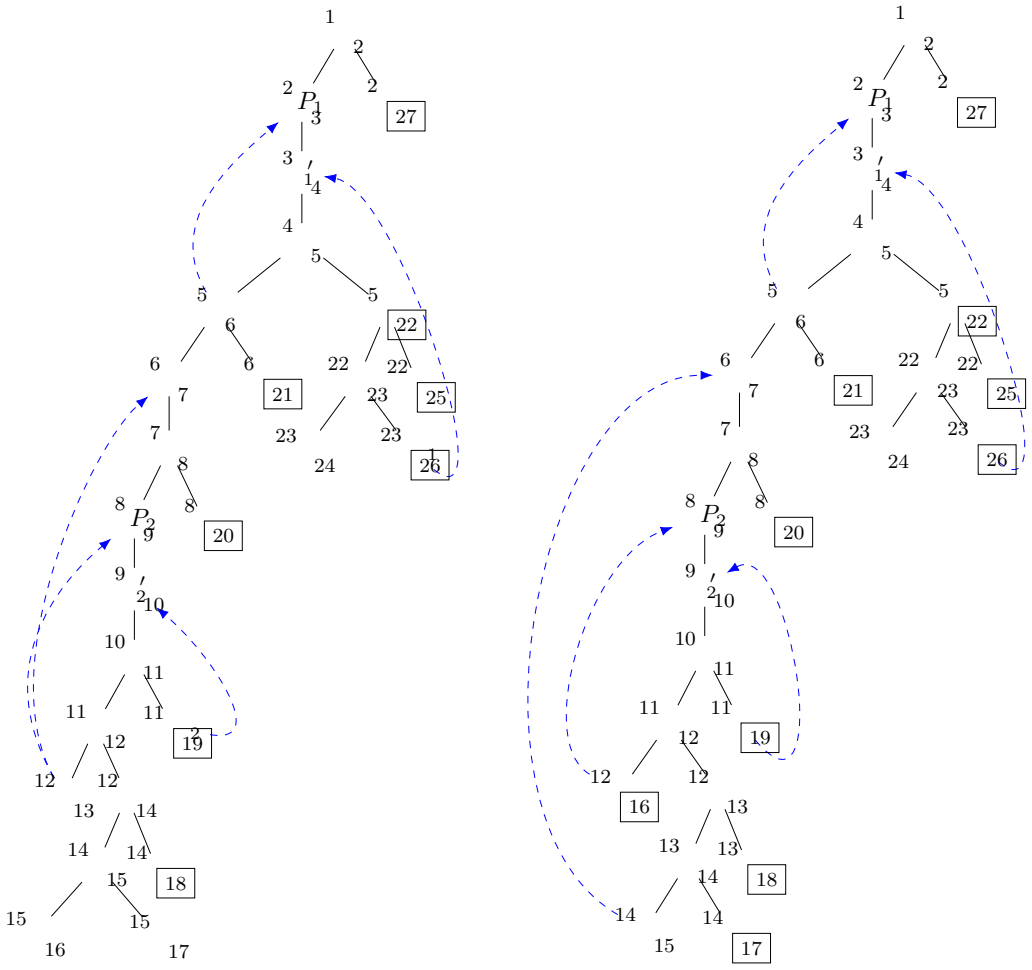


Figure 2: Annotated MG trees for the SRC and ORC sentences in Example (1) following the Head External analysis. Boxed nodes are those with tenure greater than 2.

dependency — that is, the gap within the relative clause precedes the head noun it modifies, independently of the particular RC analysis of choice.

4.1 Syntactic Assumptions

We consider two syntactic analyses proposed for Basque RCs, modeled after similar approaches commonly proposed cross-linguistically (for a summary of pre-minimalist analysis for Basque, see Gondra, 2016a)): a *Head External Analysis* (Artiagoitia, 1992, HE), and a *Head Internal Analysis* (Gondra, 2015, HI).

Head External Analysis. The HE analysis posits the presence of an RC-internal null operator coindexed with the external DP the RC modifies. This null operator raises to Spec, CP to structurally function as the head of the RC, leaving a gap in its base-generated position (Artiagoitia, 1992) .

Head Internal Analysis. According to the HI (also *Head Raising*) approach, a determiner external to the RC carrying a [+def(inite)] feature selects the relative CP. The head of the RC is a DP with a null determiner that thus moves from its base-position in the low part of the clause (either subject or object position) to Spec, CP (its landing site within the RC). Crucially, a series of "antisymmetric" movements (Kayne, 1994) is needed to ensure the correct surface word order (Gondra, 2015).

While not too distant from similar lines of RC analysis put forward for other languages (Gondra, 2016b, for an overview), Basque is characterized by a number of morpho-syntactic factors that further complicate the already generally complex approach to the analysis of RCs constructions crosslinguis-

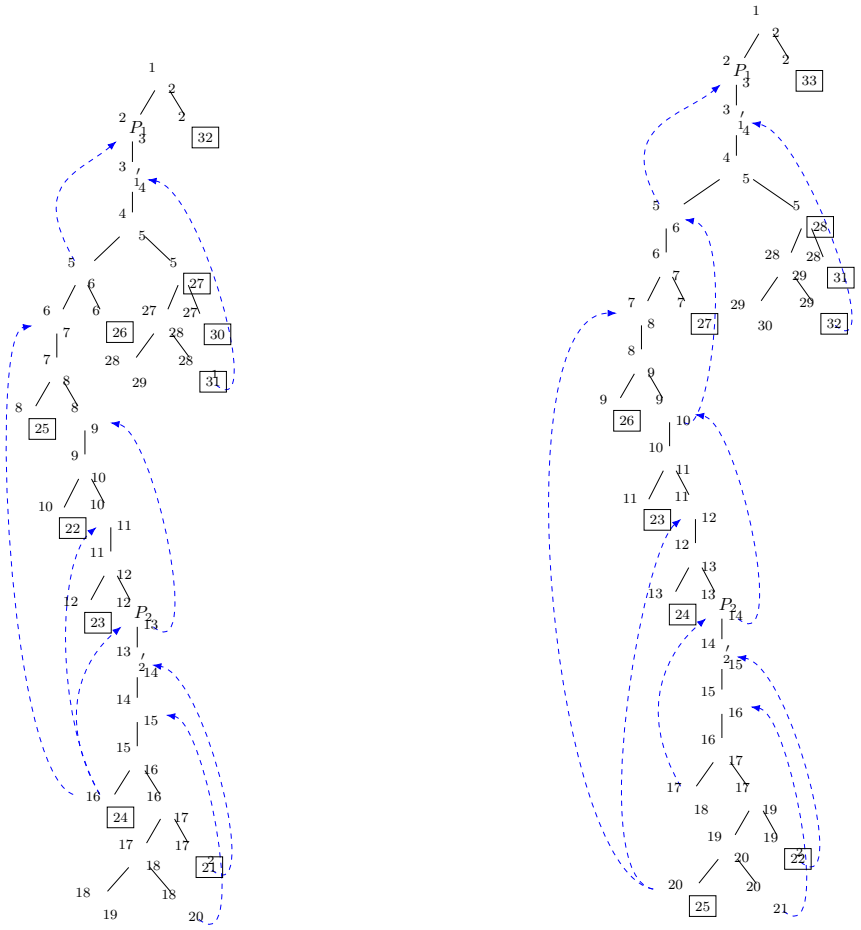


Figure 3: Annotated MG trees for the SRC and ORC sentences in Example (1) following the Head Internal analysis. Boxed nodes are those with tenure greater than 2.

tically (Bianchi, 2002a,b; Hemforth et al., 2015; Fernández, 2017, i.a.).

In particular, a number of assumptions are made in the proposed syntactic accounts in terms of functional projections and movement operations. Specifically, the HI analysis strongly relies on projections in the expanded *left periphery* (Rizzi, 1997) of the RC clause to derive the correct prenominal SOV surface order. These additional projections split the CP head into multiple projections, which encode aspectual and discourse-oriented information. Conversely, the HE analysis does not exploit these projections in the derivation, partially due to the fact that the modified noun is base-generated externally to the RC structure. It will thus be interesting to see whether and how these additional structural elements in the HI approach have any effect on the predictions made by the MG parser, when interacting with subject or object movement.

4.2 Modelling Results

As mentioned before, here we exclusively considered prenominal RCs like (1), and expect a SRC > ORC contrast. The (annotated) MG derivations for the two analyses are given in Figure 2 and Figure 3 respectively, for the prototypical sentences in (1).

With these preliminaries in place, we can look at the modelling results.[2] We evaluated the whole set of 1800 metrics defined by Graf et al. (2017) but, following previous MG parsing work, we focus our discussion on the predictions made by MAXT and SUMSIZE. As it turns out, the MG parser equipped with the ⟨MAXT,SUMSIZE⟩ metric predicts a preference for SRCs over ORCs for both analyses, but with interesting differences between the two in how this is accomplished (see Table 1). Note that for both derivations, the pairwise contrasts predicted

[2]All simulations were run with a version of the code made freely available by Graf et al. (2017) at https://github.com/CompLab-StonyBrook/mgproc.

	Head Internal			Head External		
	MaxT	**Node**	**SumSize**	**MaxT**	**Node**	**SumSize**
SRC	30	orain	50	25	orain	31
ORC	31	orain	64	25	orain	37

Table 1: Performance of MAXT and SUMSIZE for each of the RC sentences in (1), derived according to a Head Internal and a Head External analysis.

do not change whether we consider intermediate movement steps or not (cf. Zhang, 2017).

Consider again the sentences in (1). For the HE analysis, MAXT leads to a tie between SRC and ORC, with a tenure of 25 recorded on the matrix clause temporal auxiliary *orain*. Since we are considering subject RCs (i.e. the noun modified by the SRC/ORC goes to become the subject of the matrix clause) and because of the prenominal nature of the RC, every element in the matrix vP has to wait until the NP containing the RC and its noun moves to subject position in matrix Spec,TP. This results in an equivalent tenure on those nodes, given that the size (in terms of number of nodes) of the two structures is the same, independently of whether the head noun originates in subject or object position within the RC. Note that this tie is also shown on the rightmost node internal to the RC (*en*), illustrating how this is more a consequence of the prenominal RC than of having picked (consistently with the psycholinguistic literature) subject modifying structures. Interestingly, tenure on the head itself does display a subject preference. In the ORC case, *irakasleak* (in the embedded subject position) comes early in the linear sequence but has to wait for the the movement of OP from object position to Spec,CP to be resolved before it can be flushed out of the stack-memory of the parser. Nonetheless, the tie on MAXT in not a problem for a model using a ranked metric, and SUMSIZE makes in fact the correct prediction by capitalizing on the longer movement of OP in ORCs *and* on the additional movement of the embedded subject to (RC internal) Spec,TP.

Conversely, for the HI analysis, MAXT makes directly the correct prediction, registering a slightly higher tenure on the highest temporal adjunct (*orain*, but also on lower nodes) in the ORC structure. Inspecting the HI derivations more closely, we note that in this case tenure on the relativized head (*ikasleak*) predicts the opposite preference ($24 - 16$ for the SRC compared to a $25 - 20$ for the ORC). This is due to the fact that in the SRC construction *ikasleak* is predicted in Spec,vP but then it cannot be confirmed and discarded from memory until the lower VP elements (preceding it in the linear order) are found lower in the clause, and that their movement dependencies are resolved. Being predicted in object position makes it so that the waiting time for *ikasleak* is actually lower in the ORC derivation. Interestingly however, this difference disappears when we move to the higher parts of both derivations, covered by the movement of the head to Spec, ForP. The ORC derivation needing the additional movement of the RC-internal TP clause to Spec,DP is what causes its tenure to increase on the higher nodes compared to that for the SRC. SUMSIZE makes again the correct prediction by considering both these additional movement dependencies and the number of extra-projections the object head needs to move across compared to the subject head.

5 Discussion

The results above display how the MG model is able to predict a subject preference in Basque SRC/ORC constructions, in line with what reported in both production and comprehension studies (Juncal Gutierrez Mangado and José Ezeizabarrena, 2012; Munarriz et al., 2016). This success adds to previous MG modelling of sentence processing results in supporting MAXT and SUMSIZE as a combination of metrics able to capture different aspects of syntactic difficulty cross-linguistically, in ways that can give us insights into the relation between parsing and fine-grained syntactic choices.

Importantly, while the model predicts the SRC $>$ ORC ranking across two different syntactic analyses of RCs, a closer inspection reveals that it does so in strikingly different ways. In this sense, the results for the HI analysis seem mostly driven by the additional structural operations required by that analysis to derive the correct linearization. In contrast, a study of the metrics'values for the HE analysis show a higher sensitivity for the differences between subject and object RCs both in

terms of movement dependencies, and the way the tree traversal strategy of the MG parser interacts with subtler differences in the geometry of the two derivation trees. These considerations thus highlight the value of a model quantifying the relation between syntactic structure and processing load as transparently as possible, so to allow not just for quantitative predictions but also careful qualitative analyses. Specifically, this suggest ways in which this type of model could be used by both syntactitians and psycholinguists to spell out which aspects of a syntactic derivation they predict to be relevant to behavioural performance, and why.

Going back to the question of RC processing more broadly, past psycholinguistic literature has focused on well-established asymmetries between SRCs and ORCs in order to investigate the connection between universal properties of the human parser and the syntactic features of particular languages. In this sense, even though the MG model does not encode a bias towards structural locality explicitly, these results (together with previous MG modelling work on RC asymmetries in other languages) show how a subject preference could arise cross-linguistically from the interaction of language specific structural properties and generalist parsing mechanisms taking memory usage into consideration.

Finally, it is worth mentioning again that some experimental studies have reported a preference for an ORC interpretation in the processing of Basque prenominal RCs (Carreiras et al., 2010; Yetano and Laka, 2019). However, a close look at the kind of sentences tested in these studies has highlighted how the syntactic properties of Basque make prenominal SRCs temporarily ambiguous. Recall that Basque can drop several arguments (bearing ergative, absolutive, and indirect case). Additionally, the prenominal RC does not contain an explicit particle (like a wh element in English) functioning as a complementizer, which is instead attached to the subordinate verb in clause-final position (*en* in our sentences). Taken together, these characteristics make it so that a prenominal SRC could be initially interpreted as a main clause with dropped argument, at least until the parser reaches the embedded verb marked with the complementizer (Carreiras et al., 2010). Thus, past work has argued that the ORC preference found by some studies is in fact a result of the additional complexity brought in SRC structures by ambiguity resolution (Juncal Gutierrez Mangado and José Ezeizabarrena, 2012). This explanation is also in line with what has been argued for the object preference sometimes found in other head-final languages (Kwon et al., 2010, 2013; Nakamura and Miyamoto, 2013). In fact, when testing unambiguous postnominal RCs (as in (2)), Yetano and Laka (2019) report a strong preference for SRC constructions.

(2) a. Ikasle-a-ki, [zein-a-ki ei
Student-sg-ergi, [who-sg-ergi ei
irakasle-ak aipatu bait-ditu,]
teacher-pl mentioned Comp-has,]
lagun-ak ditu orain.
friend-pl has now.
'The student, who mentioned the teachers, has friends now.' **SRC**

b. Ikasle-aki, [zein-aki irakasle-a-k ei
Student-pli, [who-pli teacher-sg-erg ei
aipatu bait-ditu,] lagun-ak dira
mentioned Comp-has,] friend-pl are
orain.
now.
'The students, who mentioned the teacher, are friends now.' **ORC**

As discussed before, here we did not test our model against postnominal structures, in part due to the lack of extensive syntactic literature on the topic. Importantly though, future work in this direction will have to consider the variety of morpho-syntactic factors (especially related to case syncreticity) differentiating prenominal and postnominal constructions, and suggest fundamental ways in which the current MG model (only sensitive to tree geometry) should be expanded in order to pursue a full, in-depth investigation of syntactic processing in ergative languages. On the other hand, the preliminary results in this paper draw attention to gaps in the literature connecting theoretical syntax and psycholinguistic studies, thus showcasing once again the contribution of models like the MG parser to the broader study of the role of syntactic representation in linguistic cognition.

Acknowledgments

The authors would like to thank the audience at HSP 2023, as well as the anonymous SCiL reviewers, for valuable feedback on this work.

References

Xabier Artiagoitia. 1992. Why basque doesn't relative everything? *Anuario del Seminario de Filología Vasca" Julio de Urquijo"*, pages 11–35.

Valentina Bianchi. 2002a. Headed relative clauses in generative syntax. Part I. *Glot International*, 6.7.

Valentina Bianchi. 2002b. Headed relative clauses in generative syntax. Part II. *Glot International*, 6.8.

Joan Bresnan. 1978. A realistic transformational grammar. In Morris Halle, Joan Bresnan, and George A. Miller, editors, *Linguistic Theory and Psychological Reality*, pages 1–59. The MIT Press.

Manuel Carreiras, Jon Andoni Duñabeitia, Marta Vergara, Irene de la Cruz-Pavía, and Itziar Laka. 2010. Subject relative clauses are not universally easier to process: Evidence from basque. *Cognition*, 115(1):79–92.

Noam A. Chomsky. 1995. *The Minimalist Program*. The MIT Press, Cambridge, MA.

Rudolf P.G. de Rijk. 2007. *Standard Basque: A Progressive Grammar*. The MIT Press.

Aniello De Santo. 2020a. MG parsing as a model of gradient acceptability in syntactic islands. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 59–69.

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

Aniello De Santo. 2021a. Italian postverbal subjects from a Minimalist parsing perspective. *Lingue e linguaggio*, 20(2):199–227.

Aniello De Santo. 2021b. A Minimalist approach to facilitatory effects in stacked relative clauses. *Proceedings of the Society for Computation in Linguistics*, 4(1):1–17.

Aniello De Santo and So Young Lee. 2022. Evaluating structural economy claims in relative clause attachment. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 65–75, online. Association for Computational Linguistics.

Robert M.V. Dixon. 1994. *Ergativity*. Cambridge University Press.

Eva M. Fernández. 2017. The prosody produced by spanish-english bilinguals: A preliminary investigations and implications for sentence processing.

Lyn Frazier. 1987. Syntactic processing: evidence from dutch. *Natural Language & Linguistic Theory*, pages 519–559.

Naama Friedmann and Rama Novogrodsky. 2004. The acquisition of relative clause comprehension in hebrew: A study of sli and normal development. *Journal of Child language*, 31(3):661–681.

Sabrina Gerth. 2015. Memory limitations in sentence comprehension: a structural-based complexity metric of processing difficulty. Universitatsverlag Potsdam.

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

Edward Gibson and H-H Iris Wu. 2013. Processing chinese relative clauses in context. *Language and Cognitive Processes*, 28(1-2):125–155.

Ager Gondra. 2015. Head raising analysis and case revaluation. *Borealis : An International Journal of Hispanic Linguistics*, 4:193.

Ager Gondra. 2016a. *Basque relative clauses: Head raising, case and micro-variation within Bizkaiera*. Purdue University, PhD Dissertation.

Ager Gondra. 2016b. Universal grammar: Multiple strategies to construct relative clauses. *Estudios interlingüísticos*, (4):59–75.

Thomas Graf, Brigitta Fodor, James Monette, Gianpaul Rachiele, Aunika Warren, and Chong Zhang. 2015. A refined notion of memory usage for minimalist parsing. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 1–14, Chicago, USA. Association for Computational Linguistics.

Thomas Graf and Bradley Marcinek. 2014. Evaluating evaluation metrics for minimalist parsing. In *Proceedings of the Fifth Workshop on Cognitive Modeling and Computational Linguistics*, pages 28–36, Baltimore, Maryland, USA. Association for Computational Linguistics.

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

Henk Harkema. 2001. A characterization of minimalist languages. In *International Conference on Logical Aspects of Computational Linguistics*, pages 193–211. Springer.

Barbara Hemforth, Susana Fernandez, Charles Clifton, Lyn Frazier, Lars Konieczny, and Michael Walter. 2015. Relative clause attachment in german, english, spanish and french: Effects of position and length. *Lingua*, 166:43–64.

Tim Hunter. 2019. Left-corner parsing of Minimalist Grammars. *Minimalist Parsing*.

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.

Maria Juncal Gutierrez Mangado and María José Ezeizabarrena. 2012. Asymmetry in child comprehension and production of basque subject and object relative clauses. In *BUCLD proceedings online*.

Richard Kayne. 1994. *The antisymmetry of syntax*. MIT Press, Cambridge, MA.

Edward L. Keenan and Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8(1):63–99.

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

Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In *Model Theoretic Syntax at 10*, pages 73–82.

Nayoung Kwon, Robert Kluender, Marta Kutas, and Maria Polinsky. 2013. Subject/object processing asymmetries in korean relative clauses: Evidence from erp data. *Language*, 89(3):537.

Nayoung Kwon, Yoonhyoung Lee, Peter C Gordon, Robert Kluender, and Maria Polinsky. 2010. Cognitive and linguistic factors affecting subject/object asymmetry: An eye-tracking study of prenominal relative clauses in korean. *Language*, pages 546–582.

Elaine Lau and Nozomi Tanaka. 2021. The subject advantage in relative clauses: A review. *Glossa: a journal of general linguistics*, 6(1).

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

So Young Lee and Aniello De Santo. 2022. A computational view into the structure of attachment ambiguities in Chinese and Korean. In *Proceedings of NELS 52*, online.

Lei Liu. 2018. Minimalist parsing of heavy NP shift. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Axel Mecklinger, Herbert Schriefers, Karsten Steinhauer, and Angela D Friederici. 1995. Processing relative clauses varying on syntactic and semantic dimensions: An analysis with event-related potentials. *Memory & Cognition*, 23(4):477–494.

Jens Michaelis. 1998. Derivational minimalism is mildly context-sensitive. In *International Conference on Logical Aspects of Computational Linguistics*, pages 179–198. Springer.

George A. Miller and Noam Chomsky. 1963. Finitary models of language users. In D. Luce, editor, *Handbook of Mathematical Psychology*, pages 2–419. John Wiley & Sons.

Amaia Munarriz, Maria-Jose Ezeizabarrena, and M JUNCAL GUTIERREZ-MANGADO. 2016. Differential and selective morpho-syntactic impairment in spanish-basque bilingual aphasia. *Bilingualism: Language and Cognition*, 19(4):810–833.

Michiko Nakamura and Edson T Miyamoto. 2013. The object before subject bias and the processing of double-gap relative clauses in japanese. *Language and Cognitive Processes*, 28(3):303–334.

Alan Prince and Paul Smolensky. 2008. *Optimality Theory: Constraint Interaction in Generative Grammar*, chapter 1. John Wiley Sons, Ltd.

Owen Rambow and Aravind Joshi. 1997. A formal look at dependency grammars and phrase-structure grammars, with special consideration of word-order phenomena. *Recent trends in meaning-text theory*, 39:167–190.

Luigi Rizzi. 1997. The fine structure of the left periphery. *Elements of Grammar: Handbook in Generative Syntax*, pages 281–337.

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

Edward P. Stabler. 2011. 617 Computational Perspectives on Minimalism. In *The Oxford Handbook of Linguistic Minimalism*. Oxford University Press.

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

Chin Lung Yang, Charles A Perfetti, and Ying Liu. 2010. Sentence integration processes: An erp study of chinese sentence comprehension with relative clauses. *Brain and Language*, 112(2):85–100.

Iraia Yetano and Itziar Laka. 2019. Processing preferences in an ergative language: Evidence from basque postnominal relative clauses. *Pello Salabururi esker onez*, page 137.

Jiwon Yun, Zhong Chen, Tim Hunter, John Whitman, and John Hale. 2015. Uncertainty in processing relative clauses across East Asian languages. *Journal of East Asian Linguistics*, 24:113–148.

Chong Zhang. 2017. *Stacked relatives: their structure, processing and computation*. Ph.D. thesis, State University of New York at Stony Brook.

Modeling island effects with probabilistic tier-based strictly local grammars over trees

Charles Torres
University of California, Irvine
charlt4@uci.edu

Kenneth Hanson
Stony Brook University
kenneth.hanson@stonybrook.edu

Thomas Graf
Stony Brook University
mail@thomasgraf.net

Connor Mayer
University of California, Irvine
cjmayer@uci.edu

Abstract

We fuse two recent strands of work in subregular linguistics—probabilistic tier projections (Mayer, 2021) and tier-based perspectives on movement (Graf, 2022a)—into a probabilistic model of syntax that makes it easy to add gradience to traditional, categorical analyses from the syntactic literature. As a case study, we test this model on experimental data from Sprouse et al. (2016) for a number of island effects in English. We show that the model correctly replicates the superadditive effects and gradience that have been observed in the psycholinguistic literature.

1 Introduction

Gradience has been a long-standing issue in theoretical syntax and its interface with psycholinguistics. Is gradience a performance phenomenon or part of syntax proper? And if the latter, how could current syntactic formalisms handle gradience considering they were designed around the categorical distinction between well-formed and ill-formed structures? In this paper, we approach the issue of gradience from the perspective of *subregular linguistics*, a program equally rooted in theoretical linguistics and formal language theory. Subregular linguistics seeks to identify very restricted classes of computational (string or tree) mechanisms that can capture a wide range of linguistic phenomena. The insights from this perspective can be leveraged in a variety of ways, e.g. for new learning algorithms, novel explanations of typological gaps or linguistic universals, or to identify abstract properties that hold of both phonology and syntax.

We combine recent subregular work by Graf (2018, 2022b,a) on syntactic movement as a *tier-based strictly local* (TSL) dependency over trees with the framework in Mayer (2021) for probabilistic TSL dependencies over strings. Intuitively, a dependency is TSL iff it can be analyzed in two steps: first, one projects a tier that contains only some parts of the original structure, and second, this tier must satisfy a finite number of well-formedness constraints on adjacent structural elements. Mayer's framework allows for gradience in the string case of TSL by making this tier projection probabilistic while keeping the constraints categorical. We extend this notion of probabilistic tier projection to the the kind of TSL over trees that is used by Graf to capture syntactic movement.

The resulting framework of probabilistic TSL dependencies over trees can account for key aspects of the gradient judgments commonly observed with *island effects*, where a phrase is illicitly moved out of a containing phrase that does not allow for extraction. An example of such an island violation is shown below.

(1) a. Who does Mary say that John likes? (no island)

b. ??Who does Mary wonder whether John likes? (*whether* island)

Concretely, we test the ability of a probabilistic TSL model to handle a subset of the experimental island data in Sprouse et al. (2016).[1] The gradience observed in this experimental island data is arguably the result of many interacting factors, which may also include performance, semantics, and pragmatics (see Chaves (2022) for a recent survey). We conclude that if one wants to capture the syntactic aspects of said gradience directly in the grammar, it is eminently feasible to do so — the switch from categorical to gradient is computationally simple, natural, and does not require any modifications of the underlying syntactic analysis.

Our paper makes several contributions beyond showing the empirical viability of probabilistic TSL over trees. It continues a recent trend in subregular linguistics to increasingly unify phonology and syntax, with both aspects of language using

[1]We thank Jon Sprouse for giving us permission to use the experimental data for English from Sprouse et al. (2016).

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 155-164.
Amherst, Massachusetts, June 15-17, 2023

roughly the same kind of dependencies but applying them over strings and trees, respectively. In doing so, it also lends additional support to the specific proposals about movement in Graf (2018, 2022b,a) and gradience in Mayer (2021). The view of movement as a TSL dependency is not a mere stipulation that works in the limited case of categorical judgments, but rather provides exactly the kind of parameters that are also needed for gradience. TSL thus seems to capture a fundamental aspect of movement. Similarly, the probabilistic tier projections of Mayer (2021) have broad empirical appeal that extends far beyond the phenomena that they were originally proposed for. At the same time, our paper responds to the challenge by Chaves and Putnam (2022) to provide a TSL model of syntax that can handle gradient data. The fact that this answer requires no major changes to the categorical analysis supports the position commonly espoused by syntacticians that the issue of gradience is largely orthogonal to the enterprise of identifying the relevant syntactic structures and the operations and constraints that give rise to them.

The paper proceeds as follows. The Background section (§2) covers the relevant subregular concepts over strings. It first introduces the categorical notion of TSL (§2.1) before generalizing it to probabilistic TSL (§2.2, 2.3). We then turn to TSL over trees (§3), starting with an intuitive introduction of movement as a TSL dependency over trees and how this can be used to capture island effects in a categorical setting (§3.1–3.3). This intuition is then spelled out in formal terms (§3.4) that make it easy to combine tree TSL with the probabilistic notion of TSL from §2.3. Finally, we present the results of a modeling study (§4) showing that a simple probabilistic TSL grammar can predict many of the salient properties of the experimental data on island effects from Sprouse et al. (2016). We close with a brief discussion of the results (§5).

2 Background

This section introduces all relevant mathematical aspects of the probabilistic TSL formalism. Throughout we let Σ be an alphabet of symbols, ε the empty string, Σ^* the Kleene closure of Σ (the set of all strings of length 0 or more formed over Σ), and Σ^k the largest subset of Σ^* that contains only strings of length k. The symbols $\rtimes$ and $\ltimes$ represent left and right string boundary symbols, respectively. The : operator has type $\Sigma \to (\Sigma^* \to \Sigma^*)$ and prepends a symbol in Σ to a string in Σ^* (e.g. $a{:}bc = abc$).

2.1 Strictly local and tier-based strictly local languages

Let $s \in \Sigma^*$ for some Σ. The set of k-factors of s, $f_k(s)$, is defined as all the substrings of $\rtimes^{k-1} s \ltimes^{k-1}$ of length k. For example, $f_2(\text{tree}) = \{\rtimes\text{t}, \text{tr}, \text{re}, \text{ee}, \text{e}\ltimes\}$.

A *strictly k-local* (SL-k) grammar is a set G that contains (finitely many) forbidden substrings of length k. A string s is well-formed with respect to G iff $f_k(s) \cap G = \emptyset$, i.e. if it contains no illicit substrings of length k.[2]

Heinz et al. (2011) define a *tier-based strictly k-local* (TSL-k) grammar as a tuple $\langle G, T \rangle$ such that $T \subseteq \Sigma$ is a *tier alphabet* and $G \subseteq T^k$ is a SL-k grammar over the tier alphabet. The tier projection function π_T, which deletes from any given string all symbols not in T, is defined recursively:

$$\pi_T(\varepsilon) := \varepsilon \tag{1}$$

$$\pi_T(\sigma u) := \begin{cases} \sigma \pi_T(u), & \text{if } \sigma \in T \\ \pi_T(u), & \text{otherwise} \end{cases} \tag{2}$$

where $\sigma \in \Sigma$ and $u \in \Sigma^*$. The shape of the tier $\pi_T(s)$ projected from string s is then constrained by G exactly as in an SL grammar. Hence a string s is well-formed with respect to a TSL-k grammar $\langle G, T \rangle$ iff $f_k(\pi_T(s)) \cap G = \emptyset$.

A stringset (or equivalently, string language) is SL (TSL) iff it contains all and only those strings that are well-formed with respect to some SL-k (TSL-k) grammar, where $k \geq 0$.

2.2 Probabilistic tier projection

Probabilistic TSL (pTSL) is a generalization of TSL where π_T is a discrete probabilistic function.

A *discrete probabilistic function* $f : X \to (Y \to [0,1])$ maps pairs of strings $x \in X$ and $y \in Y$ to probabilities. These probabilities are drawn from the conditional distribution $P(y|x)$, and accordingly $\sum_{y \in Y} f(x,y) = 1$ for every $x \in X$.

Here we generalize the projection function π_T to a probabilistic version $\pi_P : \Sigma^* \to (\Sigma^* \to [0,1])$.

[2] Alternatively, a SL-k grammar can be interpreted as a collection of all well-formed substrings instead of all ill-formed substrings. In that case, string s is well-formed with respect to G iff $f_k(s)$ is a subset of G. The two interpretations are equivalent in the sense that every SL-k grammar G of forbidden k-grams generates the same set of strings as the SL-k grammar $(\Sigma \cup \{\rtimes, \ltimes\})^k - G$ of allowed k-grams.

Thus $\pi_P(x)$ returns a probability distribution over projections of some $x \in \Sigma^*$, and $\pi_P(x, y)$ returns the probability associated with projecting some $x \in \Sigma^*$ to some $y \in \Sigma^*$. It follows that $\sum_{y \in \Sigma^*} \pi_P(x, y) = 1$ for every $x \in \Sigma^*$. π_T is a special case of π_P such that the probability distribution for all $x \in \Sigma^*$ assigns a probability of 1 to a single projection.

The probabilistic tier projection π_P is calculated based on probabilities associated with the projection of each individual symbol in Σ. We define an additional function $P : \Sigma \to [0, 1]$. This function represents the probability that each symbol in Σ is projected to the tier. For example, if $P(a) = 0.7$, then there's a 70% chance the symbol a will project. We can then define π_P recursively as follows:

$$\pi_P(\varepsilon, v) := \begin{cases} 1, & \text{if } v = \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$\pi_P(\sigma_x{:}u, \varepsilon) := (1 - P(\sigma_x)) \cdot \pi_P(u, \varepsilon) \tag{4}$$

$$\begin{aligned} \pi_P(\sigma_x{:}u, \sigma_y{:}v) := {} & [\![\sigma_x = \sigma_y]\!] \cdot P(\sigma_x) \cdot \pi_P(u, v) \\ & + (1 - P(\sigma_x)) \cdot \pi_P(u, \sigma_y{:}v) \end{aligned} \tag{5}$$

where $\sigma_x, \sigma_y \in \Sigma$, $u, v \in \Sigma^*$ and $[\![\sigma_x = \sigma_y]\!]$ is an indicator function that evaluates to 1 if $\sigma_x = \sigma_y$ and 0 otherwise.

The base case (3) ensures that the only valid projection of ε is ε. In the first recursive case (4) where the input is non-empty and the projection is empty, the probability of the projection is the probability of not projecting each symbol in the input. In the second recursive case (5) where both input and projection are non-empty, we consider two possibilities for each symbol: either it projects (the first term), or it does not (the second term). The indicator variable ensures that we only consider projection as a possibility when the symbols at the beginning of the input and projection are identical.

Example. Let $\Sigma = \{a\}$ and $P(a) = 0.75$. We show that the probability of projecting $aa \to a$ is 0.375. First, by definition:

$$\begin{aligned} \pi_P(aa, a) = {} & P(a) \cdot \pi_P(a, \varepsilon) \\ & + (1 - P(a)) \cdot \pi_P(a, a) \end{aligned} \tag{6}$$

We omit the indicator variables for brevity. The first term corresponds to the case where the first a projects, and the second corresponds to the case where it does not. Solving for the two recursive instances of π_P in (6) gets us:

$$\begin{aligned} \pi_P(a, \varepsilon) &= (1 - P(a)) \cdot \pi_P(\varepsilon, \varepsilon) \\ &= 1 - P(a) \end{aligned} \tag{7}$$

$$\begin{aligned} \pi_P(a, a) = {} & P(a) \cdot \pi_P(\varepsilon, \varepsilon) \\ & + (1 - P(a)) \cdot \pi_P(\varepsilon, a) \\ = {} & P(a) \end{aligned} \tag{8}$$

Plugging these into (6) gets us:

$$\begin{aligned} \pi_P(aa, a) = {} & P(a) \cdot (1 - P(a)) \\ & + (1 - P(a)) \cdot P(a) \\ = {} & 0.75 \cdot 0.25 + 0.25 \cdot 0.75 \\ = {} & 0.375 \end{aligned} \tag{9}$$

The support of the distribution over projections, i.e. the set of projections assigned non-zero probability, is:

$$\begin{aligned} \pi_P(aa, aa) &= 0.5625 \\ \pi_P(aa, a) &= 0.375 \\ \pi_P(aa, \varepsilon) &= 0.0625 \end{aligned} \tag{10}$$

2.3 pTSL grammars

A pTSL-k grammar over an alphabet Σ is a tuple (π_P, G), where I) π_P is a probabilistic tier projection defined according to projection probabilities for each $\sigma \in \Sigma$, and II) $G \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$ is a SL-k grammar.

The function $val_{(\pi_P, G)}$ defines the probability assigned to a string u by the grammar (π_P, G):

$$val_{(\pi_P, G)}(u) = \sum_{v \in \Sigma^*} [\![f_k(v) \cap G = \emptyset]\!] \cdot \pi_P(u, v) \tag{11}$$

where $[\![f_k(v) \cap G = \emptyset]\!]$ is an indicator variable that evaluates to 0 if v contains any illicit k-factors and 1 otherwise. $val_{(\pi_P, G)}(u)$ is the sum of the probabilities of all projections of the string u that do not contain any prohibited k-factors. Note that $val_{(\pi_P, G)}$ is not a probability distribution over input strings, but rather the conditional probability of some grammatical projection given the input string.

Example. Assume the definitions of Σ and π_P from the previous example, and suppose we have a pTSL-2 grammar where $G = \{aa\}$. Then:

$$\begin{aligned} val_{(\pi_P, G)}(aa) &= \pi_P(aa, a) + \pi_P(aa, \varepsilon) \\ &= 0.4375 \end{aligned} \tag{12}$$

$\pi_P(aa, aa)$ is not included in this calculation because the projection aa contains the prohibited substring aa.

In sum, a pTSL-k grammar is the combination of a categorical SL-k grammar G with a probabilistic tier projection π_P. In contrast to the categorical tier projection π_T, π_P may project multiple tiers from any given string s. Each one of these tiers has a specific probability that is the product of the projection probabilities that resulted in this tier given s. We then sum the probabilities of all tiers projected from s that are well-formed with respect to G, yielding the conditional probability of some grammatical projection given the input s. With this understanding of how TSL over strings may be made probabilistic, we now turn to TSL over trees.

3 (p)TSL over trees

Graf (2018) generalizes TSL (more precisely the subclass TSL-2) from strings to trees. The intuition is exactly the same as in the string case: Given a tree t over alphabet Σ, we project all nodes with a label in the tier alphabet $T \subseteq \Sigma$ while preserving the ordering between those nodes in terms of dominance and precedence. SL constraints then regulate the shape of permissible tiers. A full definition of TSL-2 over trees can be found in Graf and Kostyszyn (2021), but for present purposes only the tier projection needs to be discussed in depth.

The ensuing discussion is motivated by empirical examples such as the one below, which is an instance of an *island effect*.

(2) ?? Who does Mary wonder whether John likes t?

This sentence is commonly considered degraded by native speakers of English, and syntacticians attribute this to *whether* creating an island for extraction. In the parlance of Minimalist syntax, the object *who* in the embedded clause *wh*-moves to Spec,CP of the matrix clause, but *wh*-movement is degraded out of *whether*-clauses.

Let us see, then, how this can be captured with TSL over trees using the analysis in Graf (2022a). We will first put in place feature-annotated dependency trees as a tree-based representation of the syntactic derivation (§3.1), from which we project specific tree tiers to regulate movement in a strictly local manner (§3.2). This in turn provides an easy way of modeling a wide range of island constraints as a categorical constraint against specific movement configurations (§3.3). These intuitive ideas are then made rigorous and, ultimately, probabilistic in §3.4.

3.1 Syntactic representations

Each sentence is associated with a syntactic derivation, which we represent with a dependency tree. Figure 1 gives the dependency tree for (2). Following common Minimalist assumptions, each clause consists of a verb and its three extended projections: v (which selects the subject), T (which provides the default surface position for the subject), and C (which hosts complementizers and serves as a landing site for some movement steps). Each node of the dependency tree is a lexical item, and m is a mother of a iff m selects a as an argument. If a_1 and a_2 are both daughters of m, then a_1 is a right sibling of a_2 iff a_1 is selected by m before a_2 is. That is, the right-to-left order of siblings reflects the order of selection. The geometry of the dependency tree thus encodes all relevant head-argument relations and their relative order in the derivation.

In addition, every lexical item is given a *feature annotation* inspired by the feature system of Minimalist grammars (Stabler, 1997, 2011). For each lexical item, its feature annotation encodes its category (e.g. *category feature* X^-), the categories of its arguments (e.g. the string $\mathrm{X}^+\,\mathrm{Y}^+$ of *selector features*), whether it serves as a landing site for movement steps (e.g. *licensor feature* wh^+), and whether it undergoes any movement steps (e.g. the unordered set $\{\mathrm{nom}^-, \mathrm{wh}^-\}$ of *licensee features*).[3] Note the use of capitalization to distinguish category and selector features on the one hand from licensor and licensee features on the other. All four types of features will play a key role in deciding which nodes should be projected onto a given tier.

3.2 Movement tiers

With the basics of feature-annotated dependency trees in place, we turn to tier projection for movement. In Fig. 1, we have three separate movement steps: two instances of subject movement, and one instance of wh-movement. Let us consider the former first. The subject *Mary* in the matrix clause moves to Spec,TP of the matrix clause, and the subject *John* in the embedded clause moves to Spec,TP of the embedded clause. In both cases, this is implicitly encoded by the fact that *Mary* and *John* carry the licensee feature nom^-, and the

[3] In contrast to Minimalist grammars, licensee features are unordered in our system so that a mover with multiple licensee features will always target the closest dominating nodes with matching licensor features. This affects neither weak nor strong generative capacity (Graf et al., 2016) but is a crucial prerequisite for capturing movement dependencies via tiers.

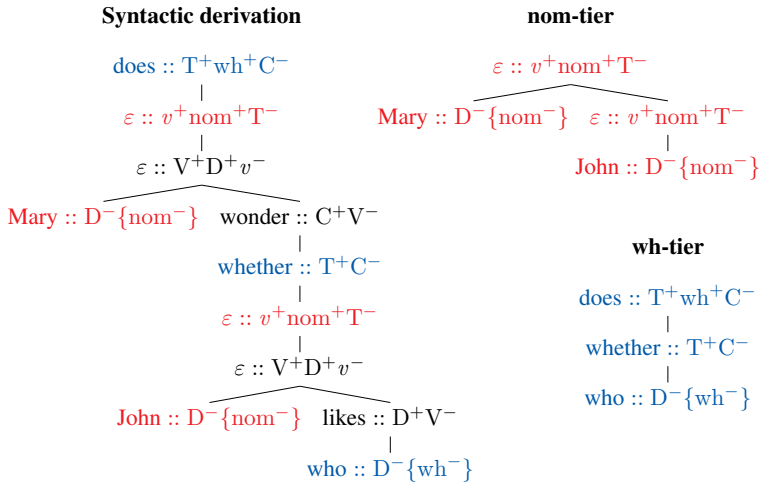


Figure 1: Syntactic derivation for (2), its well-formed nom-tier (in red), and the ill-formed wh-tier (in blue)

corresponding T-heads carry the matching licensor feature nom^+. A lexical item with some licensee feature f^- will always move to a specifier of the closest dominating lexical item with matching licensor feature f^+. This is why *Mary* moves to Spec,TP of the matrix clause, whereas *John* moves to Spec,TP of the embedded clause. Each one of these subject movement steps is well-formed, and as pointed out by Graf (2018), this can be verified in a tier-based strictly local manner.

In order to determine whether the derivation contains any illicit instances of subject movement, we construct a subject movement tier that contains only nodes that matter for subject movement. At the very least, this tier must contain every lexical item that carries nom^+ or nom^- (as we will see during the discussion of wh-movement, projecting additional lexical items is exactly what gives rise to island effects). The resulting *nom-tier* is shown in Fig. 1. Note how the dominance relations in the tier match the dominance relations in the dependency tree. Moreover, *Mary* is the left sibling of the embedded T-head on the tier because in the dependency tree, *Mary* precedes the embedded T-head (that is to say, *Mary* is reflexively dominated by a node that is the left sibling of a node that reflexively dominates the embedded T-head). The nom-tier is well-formed iff it obeys both of the following conditions for movement tiers:

(3) **Well-formedness of an f-tier**

a. Every node with f^+ has exactly one node with f^- among its f-tier daughters.

b. Every node with f^- has an f-tier mother that carries f^+.

Both of these conditions are met in the nom-tier, which entails that all subject movement steps in the derivation are well-formed.

3.3 Categorical island effects

Now consider the case of wh-movement of *who* from the embedded object position to Spec,CP of the matrix clause. Without additional assumptions, this movement step should be well-formed. If we construct the corresponding wh-tier, it consists only of *does* with *who* as its only daughter. As the former carries wh^+ and the latter wh^-, the conditions in (3) are met and the tier should be well-formed. But we already saw that (2) is not considered well-formed due to the presence of *whether*. Suppose, then, that we also project *whether* onto the wh-tier, yielding the wh-tier in Fig. 1. Both conditions in (3) are now violated by the wh-tier because *whether* intervenes between *does* and *who*. Island effect thus arise whenever an element that does not carry the relevant features is projected onto a tier and destroys the mother-daughter configuration between a mover and its target.[4]

This same idea can be used to capture other island effects. In addition to the *whether island*

[4]Note that projecting *whether* on the nom-tier would not destroy any such configurations. Irrespective of whether one projects *whether*, the nom-tier is well-formed. In general, it is safe to assume that islands project onto all movement tiers, unless there is good empirical evidence that a specific movement type is not subject to a specific island condition. For the purposes of this paper, what exactly projects onto the nom-tier does not matter as all our modeling will focus exclusively on the wh-tier.

constraint described above, we will also examine the *adjunct island constraint* and the *complex NP constraint*. The adjunct island constraint prevents extraction from adjuncts, e.g. *because*-clauses as in (4a). The complex NP constraint prevents extraction from sentential complements of nouns (4b). Both effects also arise with extraction from relative clauses as in (4c) and (4d), respectively. For simplicity, we will conflate the difference between wh-movement and relative clause extraction and treat both as involving the features wh^+ and wh^- for the rest of this paper.

(4) a. * Who did Mary complain because John likes t?

b. * Who did Mary deny the rumor that John likes t?

c. * I saw the congressman who Mary worries if John respects t.

d. * I saw the man who Mary heard the rumor that John likes t.

All these cases can be analyzed as some lexical item projecting onto a movement tier and disrupting the local licensing relations there. The adjunct island constraints are captured by projecting the heads of adjunct islands, for example *because* and *if*. The complex NP constraint amounts to projecting all nouns that select a CP as their only argument (i.e. every lexical item whose feature annotation contains the substring $\mathrm{C}^+\mathrm{N}^-$). Crucially, the decision to project a lexical item only requires maximally local information: the surface realization of the lexical item and/or its feature annotation.[5]

However, all these accounts are hamstrung by the fact that tiers are either well-formed or ill-formed. It is not possible to express the fact that, say, *whether*-island violations are not judged as degraded as extraction from *because*-clauses. One easy way to add gradience to this system is to adapt the probabilistic tier projection mechanism of Mayer (2021), which we discussed in §2.2 and §2.3.

3.4 Probabilistic tree tier projection

In order to define a probabilistic tier projection for trees, we first need a rigorous definition of categorical tier projection for trees. We adopt the logic-based definition of Graf and Kostyszyn (2021) where a tier is just the result of enriching the dependency tree with relations for *tier daughter* and *tier sibling*.

Let us use $\lhd^+$ ($\lhd^*$) to denote proper (reflexive) dominance in the dependency tree, i.e. $x \lhd^+ y$ ($x \lhd^* y$) holds in dependency tree t iff x properly (reflexively) dominates y in t. We also use $x \prec y$ to denote that x is a left sibling of y in t. Furthermore, the predicate $T(x)$ is true iff the label of x (e.g. wonder :: $\mathrm{C}^+\mathrm{V}^-$ in Fig. 2) is part of our tier alphabet T. We define proper dominance on tier T ($\lhd_T^+$) and use that to subsequently define the daughter-of relation over tier T ($\lhd_T$), which in turn is needed to define the left-sibling relation over tier T ($\prec_T$):

$$
\begin{aligned}
x \lhd_T^+ y \Leftrightarrow & T(x) \wedge T(y) \wedge x \lhd^+ y \\
x \lhd_T y \Leftrightarrow & x \lhd_T^+ y \wedge \neg\exists z[x \lhd_T^+ z \wedge z \lhd_T^+ y] \\
x \prec_T y \Leftrightarrow & \exists z[z \lhd_T x \wedge z \lhd_T y] \wedge \\
& \exists z, z'[z \lhd^* x \wedge z' \lhd^* y \wedge z \prec z']
\end{aligned}
$$

These predicates implicitly define the tier T over dependency tree t and provide the relevant structural relations for tier constraints such as the one-to-one match between mothers with licensor features and daughters with licensee features we encountered in (3).

In order to turn this categorical notion of tree tiers into a probabilistic one, it suffices to make membership in the tier alphabet probabilistic. For example, if elements with the same label as x have a probability of 0.7 to project onto tier T, then the predicate $T(x)$ has a probability of 0.7 of being true. This is the only required change. The definitions of $\lhd_T^+$, $\lhd_T$, and $\prec_T$ remain exactly the same—it is only the interpretation of $T(x)$ that becomes probabilistic. Once this change is made, the probability of a given tier projection is calculated in exactly the same manner as in the string case (§2.2): it is the product of $T(x)$ for every x that projects, and $(1 - T(x))$ for every x that does not project. The overall conditional probability of a given tree having some grammatical projection is

[5] Mathematically, the tier projection may use any information that can be encoded in terms of a finitary annotation scheme for lexical items. This includes, among other things, the semantic denotation of the lexical item, a higher-dimensional vector representation derived from word embeddings, aspects of information structure such as topic and focus, or basic frequency information in terms of a finite classification system like *very rare*/*rare*/*common*/*ubiquitous*. Any kind of annotation that preserves Minimalist grammars' requirement that the set of lexical items must be finite is mathematically permissible. So even though we will limit ourselves to purely syntactic information in our subsequent discussion of island effects, the approach could be extended to consider at least some of the semantic and pragmatic factors observed in Chaves (2022) and the studies referenced therein.

also calculated in the same manner as the string case: it is the sum of the probabilities of all its possible licit tier projections.

4 Modeling study

The next section presents a computational modeling study where a simple pTSL grammar over trees is fit to experimental data on English island effects from Sprouse et al. (2016).[6] We demonstrate that in addition to exhibiting the superadditive effects found by Sprouse et al., it can also represent the gradience observed across judgments of different island effects.

4.1 Methods

The stimuli from Sprouse et al. (2016) were given a syntactic analysis using feature-annotated dependency trees as described in §3. We restricted ourselves to the subset of sentences exhibiting the island effects described above: *whether* islands, adjunct islands, and complex NP islands. We also omitted filler sentences. This produced a total of 160 trees.

Sprouse et al. (2016) partitions the data within each island effect type based on two factors: whether the sentence contains an island structure, and whether the node that undergoes movement is located in the matrix clause or the embedded clause. Examples of the four combinations of these two factors are show in (5) for *whether* islands (from Sprouse et al., 2016).

(5) a. Who t thinks [that John bought a car]? (non-island, matrix clause)

b. What do you think [that John bought t]? (non-island, embedded clause)

c. Who t wonders [whether John bought a car]? (island, matrix clause)

d. What do you wonder [whether John bought t]? (island, embedded)

This factorial design is intended to separate the effects of extracting from a matrix clause vs. extracting from an embedded clause, and also the effects of the presence or absence of an island structure. In particular, Sprouse et al. expect that sentences like (5d), which are the only ones that contain syntactic island configurations, should display *superadditive effects*. That is, the effect of these configurations on human judgments should be greater than the independent contributions of extracting out of an embedded clause, as in (5b), and the presence of an island structure that is not extracted over, as in (5c).

The dataset from Sprouse et al. (2016) contains about 14 Likert scale ratings for each sentence we considered. Because our model is unable to represent cross-speaker variability in judgments, we assigned each sentence the mean rating across participants. Following Sprouse et al. (2016), we use ratings that were Z-score normalized by participant rather than the raw Likert scores.

Using the dependency trees and Z-scores, we fit a pTSL grammar to the data by finding the optimal projection probabilities: that is, those that align as closely as possible the scores assigned by the model to the scores assigned by humans. We do this by first transforming the mean Z-score values to fall in the range $[0, 1]$ and then minimizing the mean squared error between the transformed human acceptability judgments and the probabilities assigned by the model. This minimization was performed using `scipy.optimize.minimize` (Virtanen et al., 2020) with bounded L-BFGS optimization to ensure each projection probability is within the interval $[0, 1]$.

We *a priori* fixed most of the projection probabilities to 0 (irrelevant nodes) or 1 (nodes with wh^+ or wh^-), and we fit only projection probabilities for nodes that could feasibly induce island effects. This was done to facilitate interpretability of the model, speed up the model training, and offset the comparatively small size of the training set.

The nodes whose projection probabilities were fitted were:

- that :: $\mathrm{T}^+\mathrm{C}^-$
- whether :: $\mathrm{T}^+\mathrm{C}^-$
- if :: $\mathrm{T}^+\mathrm{C}^-$
- all nodes whose feature annotation contains the substring $\mathrm{C}^+\mathrm{N}^-$

The first three items are potential blockers for the whether island and adjunct island constraints; nodes with $\mathrm{C}^+\mathrm{N}^-$ correspond to nouns that head complex NPs and should thus induce complex NP island effects. There are a set of seven nouns in the data that have this featurization (*rumor*, *claim*, etc.). For simplicity we assume all such nouns have the same projection probability and treat this as a single parameter.

Finally, nodes representing *wh*-movers and landing sites were set to always project. The latter

[6] The code and data can be found at: https://github.com/connormayer/pTreeTSL

includes interrogative C-heads and relative clause C-heads with feature string $\mathrm{T}^{+}\mathrm{wh}^{+}\mathrm{C}^{-}$ (recall that we use wh both for wh-movement and for relative clause movement). The former consists of wh-pronouns with the feature string $\mathrm{D}^{-}\{\mathrm{wh}^{-}\}$).

Because fitting the model is stochastic, we performed training ten times in order to determine whether the probabilities reliably converge to the same values.

4.2 Results

For our four features of interest, learned projection probabilities showed little variance across the ten runs. Each converged within 10^{-3} to the same projection probability. This aligns with the suggestion in Mayer (2021) that the optimization function when fitting a pTSL model in this way is concave. We report projection probabilities and scores averaged across the ten runs.

Table 1 shows the projection probabilities learned by the model for the four nodes of interest. Recall that higher projection probabilities increase the likelihood of these nodes projecting to the wh-tier and intervening between a tier mother with wh^{+} and its tier daughter with wh^{-}. Therefore, higher projection probabilities should correspond to lower ratings for the relevant island structures. The relative projection probabilities show that *if* is mostly likely to act as a blocker, *that* is least likely, and complex NPs and *whether* are intermediate between the other two.

The mean human scores and the mean model scores for each sentence type are shown in Fig. 2. The model scores capture several important aspects of the human judgments: (a) extracting out of a matrix clause is uniformly judged to be better than extracting out of an embedded clause; (b) extracting out of an embedded clause over an island produces the expected superadditive effects; and (c) the relative badness of the five types of island extraction (the right point in the red lines in Fig. 2) matches the relative badness reflected in the human judgments.

Node	Projection probability
that :: $\mathrm{T}^{+}\mathrm{C}^{-}$	.46
$\mathrm{C}^{+}\ \mathrm{N}^{-}$	.63
whether :: $\mathrm{T}^{+}\mathrm{C}^{-}$	.73
if :: $\mathrm{T}^{+}\mathrm{C}^{-}$	.89

Table 1: Mean projection probabilities

There are a number of aspects of the data the model fails to capture. First, it over-predicts this superadditivity in the case of relative clause adjunct islands, where it was not found in the human data. Second, it does a poor job of predicting the relative badness of forms in the matrix extraction condition. These sentences are not ungrammatical in terms of the wh-tier, and the model accordingly assigns them all probabilities of 1 in these cases. In particular, humans generally assign worse scores when an island structure is present, even if it is not a blocker, while the model cannot do so. Finally, although it captures the general tendency for extraction out of embedded clauses to be worse than extraction out of a matrix clause, the relative effect of this in different island types is not captured by the model.

5 Discussion

Assessing the performance of the probabilistic TSL model for the English island data from Sprouse et al. (2016) is a sutble affair because there are so many factors that could influence what Likert scores participants assign to specific stimuli. Syntactic constraints, processing difficulties, lexical frequency, semantics, pragmatics, and information structure may all be involved. By limiting our attention to only the phonetic exponents of lexical items and their feature make-up, we are asking the model to capture the experimental data as well as possible with only syntactic information. In that respect, the model succeeds as it gives rise to super-additivity, which has been argued to be the primary reflex of syntax in experimental island effect data.

Admittedly the model does not do a perfect job, and future work is needed to fully explore these issues. For example, the model overpredicts superadditivity in relative clause adjunct islands. This raises the question whether alternative analyses of relative clauses would have fared better in this respect, and if not, what non-syntactic factors could explain the less pronounced nature of superaddivity in these constructions. Similarly, although the model is able to capture superadditivity and the relative badness of the types of island violations considered here, it does poorly in predicting the variability in judgments of the non-island cases and the short forms of island cases. Once again this might indicate the need for a revised syntactic analysis, or point towards non-syntactic factors.

Crucially, these non-syntactic factors are not necessarily beyond the purview of the pTSL model —

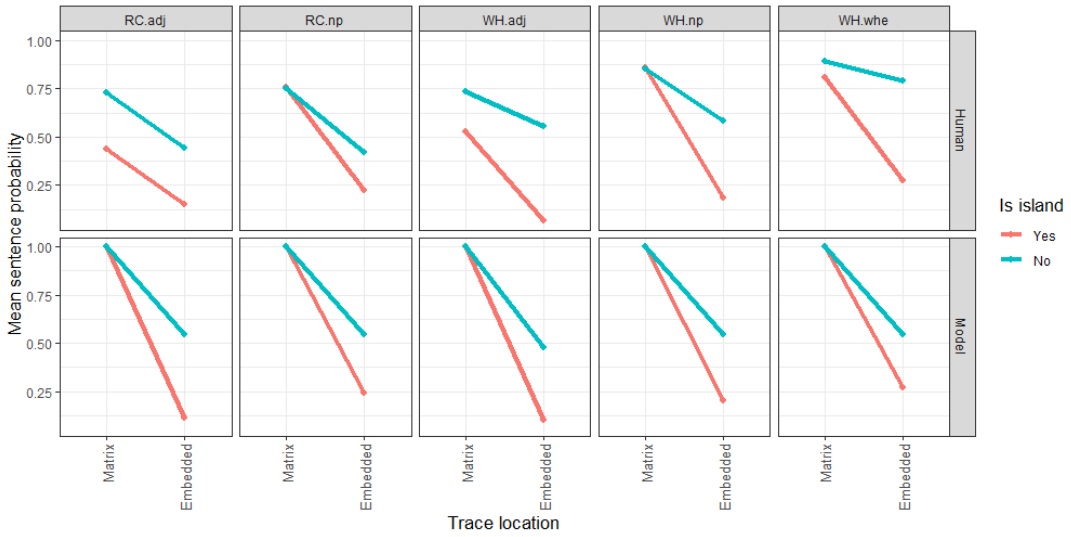


Figure 2: Human judgments from Sprouse et al. (2016) (top) and mean model judgments after training (bottom)

any information that can be lexicalized can be taken into account by the tier projection function. For example, an analysis that encodes topic and focus as movement of phrases to specific syntactic positions furnishes specific movement features that encode the topic-focus distinction and thus could serve as parameters for the tier projection and the constraints that apply on tiers. Hence it is important not to equate the syntax-only approach we took in this paper with the limits of what can be modeled with pTSL.

In relation to this, it is also important to remember that the probabilities themselves might encode remnants of non-syntactic factors and hence don't give us a "pure" picture of the role of syntax in island effects. It is likely that the learned projection probabilities shown in Table 1 encode some effects related to processing rather than syntax. In particular, *that* has a relatively high projection probability despite it not being considered a syntactic blocker. The model has likely assigned this probability in order to encode the decrease in acceptability between extraction out of a matrix clause and extraction out of an embedded clause. Integration of the model proposed here with other models, e.g. the processing approach of De Santo (2020) to gradience in adjunct islands, has the potential to shed more light on whether effects such as this should be modeled as part of the grammar.

Our primary goal was to show that the switch from categorical TSL (and the categorical syntactic analyses that can be expressed this way) to a probabilistic, gradient model is easy and empirically viable. The task of adequately modeling island effects with pTSL is much larger than this, but we are confident that pTSL will be able to provide novel insights in this domain.[7]

6 Conclusion

We have presented pTSL as a simple probabilistic extension of TSL syntax that makes it easy to add gradience to existing syntactic analyses (provided they can be stated in terms of categorical TSL). The key idea of this extension is the switch to a probabilistic tier projection function. We discussed island effects as an example of the empirical viability of this approach: the combination of a standard Minimalist analysis with probabilistic tier projection is able to replicate the superadditive effects of extraction out of islands and the gradience in the relative badness of different types of island constructions.

Acknowledgements

The work carried out by Kenneth Hanson and Thomas Graf for this project was supported by the National Science Foundation under Grant No. BCS-1845344.

[7] To avoid potential confusion: our pTSL model is not intended to be a model of how island effects are learned. The model is trained on scores assigned to the data in experimental contexts, which is not a realistic learning scenario. But the model is of interest for learning because its parameters are interpretable and it does successfully encode syntactic contributions to judgments of island effects, including superadditivity and gradience.

References

Rui P. Chaves. 2022. Sources of discreteness and gradience of island effects. *Languages*, 7:245.

Rui P. Chaves and Michael T. Putnam. 2022. Islands, expressiveness, and the theory/formalism confusion. *Theoretical Linguistics*, 48(3–4):219–231.

Aniello De Santo. 2020. MG parsing as a model of gradient acceptability in syntactic islands. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 59–69. Association for Computational Linguistics.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. *Proceedings of CLS*, 53:117–136.

Thomas Graf. 2022a. Subregular linguistics: Bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48:145–184.

Thomas Graf. 2022b. Typological implications of tier-based strictly local movement. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 184–193, online. Association for Computational Linguistics.

Thomas Graf, Alëna Aksënova, and Aniello De Santo. 2016. A single movement normal form for Minimalist grammars. In *Formal Grammar: 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016*, pages 200–215, Berlin, Heidelberg. Springer.

Thomas Graf and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist Grammars. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 275–285, Online. Association for Computational Linguistics.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Connor Mayer. 2021. Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 39–50, Online. Association for Computational Linguistics.

Jon Sprouse, Ivano Caponigro, Ciro Greco, and Carlo Cecchetto. 2016. Experimental syntax and the variation of island effects in English and Italian. *Natural Language & Linguistic Theory*, 34(1):307–344.

Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.

Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272.

Morpheme combinatorics of compound words through Box Embeddings

Eric Rosen
(formerly at) University of Leipzig
errosen@mail.ubc.ca

Abstract

In this study I probe the combinatoric properties of Japanese morphemes that participate in compounding. By representing morphemes through box embeddings (Vilnis et al., 2018; Patel et al., 2020; Li et al., 2019), a model learns preferences for one morpheme to combine with another in two-member compounds. These learned preferences are represented by the degree to which the box-hyperrectangles for two morphemes overlap in representational space. After learning, these representations are applied to test how well they encode a speaker's knowledge of the properties of each morpheme that predict the plausibility of novel compounds in which they could occur.

1 Introduction

In Japanese, compounding is very productive, particularly when the morphemes involved have a Sino-Japanese reading. The NHK Pronunciation Dictionary (NHK, 2016) lists 26,867 two-member compounds, in which 2,901 different morphemes occur as morpheme 1 and 2,740 morphemes occur as morpheme 2. The compounds are listed with their kanji characters, followed by their pronunciation in Japanese *katakana* syllabic characters. Following Nagano and Shimada (2014), I adopt the hypothesis that Japanese kanji characters correspond with morphemes, even when the pronunciation of the character might differ in different contexts or when there is what Nagano and Shimada (2014) refer to as a 'dual reading' for a character, with one '*kun*' reading as a native Yamato Japanese word and the other '*on*' reading with an unrelated Sino-Japanese pronunciation borrowed from Chinese.[1] An example is 作, 'make', whose Sino-Japanese pronunciation is *saku* and which occurs as a native Yamato morpheme in verb *tukuru* 作る 'make'. It occurs as the first member of 28 two-member compounds listed in NHK (2016) of which eight examples are shown in Table 1.

作品 *saku* + *hin* 'goods' = 'production'
作家 *saku* + *ka* 'house' = 'author' (*sak-ka*)
作成 *saku* + *see* 'become' = 'to make'
作戦 *saku* + *sen* 'battle' = 'strategy'
作文 *saku* + *bun* 'sentence' = 'composition'
作曲 *saku* + *kyoku* 'music' = 'composed music'
作業 *saku* + *gyoo* 'business' = 'work' (*sa-gyoo*)
作者 *saku* + *sya* 'person' = 'author'

Table 1: Eight compounds beginning with *saku*, 作, 'make'

For many of these compounds, the combination of morpheme 1 with morpheme 2 is transparently compositional. For example, the meaning 'production' of the first example in Table 1 follows logically from 'make' + 'goods'. But many other compounds such as 親切 *sin-setu* 'kindness', formed from 親 *sin* 'parent' and 切 *setu* 'cut', are not compositional in any obvious way, unless the constituent morphemes are taken to be polysemous, with sub-meanings that do in fact compose to denote, in this example, 'kindness'.[2] The question I tackle here is, what kinds of representations of morphemes might a speaker have that enables them to predict whether morphemes can combine in a compound word? Especially in cases where a morpheme is bound, and thus never occurs in isolation, deduction of its contribution to compounds it occurs in becomes a matter of its relation to other morphemes it combines with rather than some meaning that it may or may not have on its own.[3]

[1] As shown in Nagano and Shimada (2014), morphemes or combinations of them in a Sino-Japanese vs. a native Yamato reading occur in complementary grammatical contexts: "[A] *kanji* graph loses its dual pronunciation once it is given a grammatical context." (p. 331)

[2] Nelson (1987) lists 'intimate, familiar, friendly' and 'kind' among many sub-meanings for the two kanji characters, respectively.

[3] As elaborated on by Nagano and Shimada (2014), a Sino-Japanese reading of a morpheme, whether it also has an additional Yamato reading or not, generally acts like a bound

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 165-174.
Amherst, Massachusetts, June 15-17, 2023

I thus investigate how learned representations of morphemes can predict how plausible their combination would be in a compound word.

The paper is organized as follows. In §2 I introduce Box Embeddings, which I use to represent morphemes in compounds. In §3 I describe a model trained to learn Box Embeddings of Japanese morphemes that occur in compound words that is based on which morphemes occur or do not occur together. In §4 I give details of the model's method of training. In §5 I discuss what the trained model predicts about hypothetical compounds that were not seen in training. In §5.1 I probe further into the kinds of associations between morphemes that the model finds in training. In §6 I show graphically what some examples of overlap of box embeddings look like in two dimensions. In §7 I discuss the issue of morpheme frequency and to what extent it can be an indicator of how perspicuously two morphemes can combine in a compound. In §8 I address the question of exactly what the model is learning about the morphemes it is trained on. In §9 I present results of some further testing of hypothetical compounds that the model predicts to have the most ideal choice of a morpheme 2 to combine with each of the morpheme 1s in the corpus. In §10 I compare the box embedding model with a model trained on simple vector embeddings. In §11 I conclude with a discussion of what the next steps would be in continuing the current investigations.

2 Box Embeddings

In a manner analogous to the way that word embeddings are based on the context in which a word is found (e.g., 'Word2vec', Mikolov et al. (2013)), here I represent morphemes that occur in Japanese compounds according to what other morphemes they occur with in compounds. To do so, I use Box Embeddings (Vilnis et al., 2018), which represent entities as hyperrectangles in a space of n dimensions. A box is defined in Chheda et al. (2021) as the Cartesian product of closed intervals and can also be defined by z_i and Z_i, the minimum and maximum coordinates of the box in each dimension i. Box embeddings have advantages over simple vector representations. As discussed by Vilnis et al. (2018), the relation between two hyperrectangles is asymmetric, unlike the relation between two vectors. As shown in two dimensions in Figure 1, the

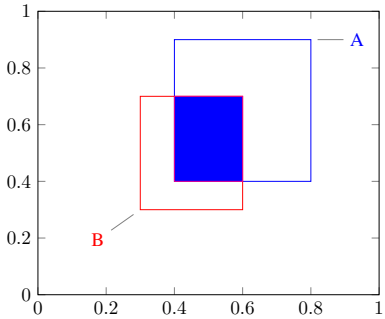


Figure 1: Overlap of boxes A and B in two dimensions

degree to which element A entails element B can be expressed as the amount of the volume of the box or hyperrectangle representing A that occurs in the box representing B, or $\frac{\text{vol}(A \cap B)}{\text{vol}(B)}$.

This differs from the degree to which B entails A. Applying this to two members of a bimorphemic compound $M_1 M_2$, we can distinguish the degree to which M_1 as a first member entails the occurrence of M_2 as the second member from the opposite implication. We could also combine the two implications for prediction by averaging the implication from M_1 to M_2 with the implication from M_2 to M_1.

A second advantage of box embeddings given by Vilnis et al. (2018) is that they can capture negative correlations between concepts. In our case, for reasons given in §3, we want to train a model based not just on which M_2 can occur with each M_1 in a compound and vice-versa, but also on which choices of M_2 do not occur with M_1.

See Vilnis et al. (2018); Patel et al. (2020); Li et al. (2019) for detailed discussion of the theoretical basis of box models and how they compare to other related models that are based on geometric structures.

3 The task

Our task is to train a model to learn box embeddings of morphemes that occur in two-member Japanese compounds based on (a) combinations of morphemes that occur together and (b) random pairs of morphemes that do not occur together in any compound. The training objective for occurring combinations is to have their box embeddings overlap as much as possible – that is for $\frac{\text{vol}(M_1 \cap M_2)}{\text{vol}(M_1)}$ and $\frac{\text{vol}(M_1 \cap M_2)}{\text{vol}(M_2)}$ to each approach 1.0. In the case of non-occurring combinations, we want these intersecting volumes to approach 0. I use the corpus that

morpheme that only occurs in combination with another morpheme in a compound.

was described on page 1 of 26,867 two-member compounds extracted from NHK (2016) in which 2,901 different morphemes occur as M_1 and 2,740 morphemes occur as M_2. Morphemes are represented orthographically by Japanese kanji characters as in the examples in Table 1. Training on randomly-chosen non-occurring combinations of morphemes in an equal number to occurring combinations is essential to prevent the box embeddings of morphemes from expanding to the extent that all the morphemes would coincide in the representational space. If this were to happen, the model would incorrectly predict that every M_1 can occur with every M_2 and vice-versa. Because there are $2,901 \times 2,740$ or almost 8 million possible combinations of M_1 and M_2, in 10 million data points of training, we expect each hypothetical combination to be trained on only slightly more than once on average, whereas each existing compound will have been trained on in each implicational direction 10^7 (number of updates) $\div 26,867$ (number of compounds) or about 372 times on average. This means that even though one non-occurring combination was trained on once for every occurring combination, in testing, a randomly chosen non-occurring combination of M_1 and M_2 will have had negligible or no training based on that particular combination as a non-occurring compound (which would seek to make their boxes disjoint); rather the volume overlap for that combination that appears in testing will have resulted from training more generally on what other morphemes each of that M_1 and M_2 occur or do not occur with.

On the hypothesis that these learned embeddings of morphemes inform how well they combine with each other to form a compound, the greater the intersecting volume ratio of two embeddings, the more we expect their combination to be plausible.

4 Training

Using the Pytorch implementation of the open-source library for box embeddings in Chheda et al. (2021), I trained the corpus data with an embedding dimension of 16, a learning rate of 0.01 and a mean squared error loss function. On each of 10 million updates, a randomly chosen occurring compound and a random combination of morphemes that do not occur were each chosen. For each, prediction of M_2 from M_1 and M_1 from M_2 are made separately. If a morpheme can occur both as an M_1 and an M_2, it is given a separate embedding for each. The loss is the squared difference between the volume overlap ratio and 1.0 for existing compounds and 0.0 for non-occurring compounds.

After training, the volume overlap ratio scores for the actual compounds vary from low scores of 0.122 (M_2 from M_1) and 0.111 (M_1 from M_2) up to high scores above 0.99. We find that the compounds whose score comes close to 1.0 in training tend to be compounds for which each member occurs in no other compounds in the database. This means that in training, the embeddings for each member will be drawn to overlap with each other without any countervailing forces pulling them away because of an occurrence with other morphemes. Their training on non-occurring compounds will have pulled their embeddings in randomly diverse directions whose net effects should cancel out and thus not move the two embeddings away from each other. On the other hand, compounds with low scores will tend to have at least one member that occurs in many compounds. An example is 上巳 *zyoo-si*, 'March 3rd dolls festival' (lit. 'upper' + 'sixth sign in the Chinese zodiac'), with a relatively low score of 0.124 for predicting M_2 from M_1. M_1 上 *zyoo* 'upper' occurs as the first member of 90 other two-member compounds whereas 巳 *si* occurs in only one other compound. The other 90 morphemes that combine with M_1 上 *zyoo* will pull its embedding in a very different direction from where 巳 *si* pulls it, given the idiosyncratic meaning of this apparently atypical combination of morphemes. The average scores for real compounds are 0.652 for predicting M_2 from M_1 and 0.651 for predicting in the other direction.

For randomly chosen compounds in which a different morpheme was substituted for either M_1 or M_2, one compound scores above 0.9 for predicting M_2 from M_1. 心物 'heart" + 'thing' scores 0.927 predicting M_1 from M_2. 心 occurs in 46 compounds and 物 in 152. Not only does 物 *butu, motu*[4] 'thing' combine as a M_2 with many M_1s, but 心 'heart' has an existing compound 心事 *sin-zi* with morpheme 事 which also means 'thing' but with a more abstract meaning than 物. Moreover, there are 24 M_1s that form compounds with both 物 and 事, one example with M_1 変 *hen* 'disaster; strange' being 変物 *hen-butu* 'eccentric person' and 変事 *hen-zi* 'accident, disaster'. This example illustrates the way that the model can capture par-

[4] In some compounds, 物 has the native Yamato reading *mono*

allel analogies. Here, it predicted the possibility of a compound A+X if there exists a compound B+X and there exist many pairs of compounds of the form (A+Y, B+Y). Although 心物[5] is not listed in NHK (2016), an internet search finds it occurring on a page of Japanese text at University of Virginia Library.

At the opposite end of the scale, for 刎頚 *hun-kee* 'decapitate + neck = decapitation', when 呵 *ka* 'scold' is substituted for M_2, the score is 0.061. All the morphemes involved participate in only one compound in the corpus. For the real compound, the score is high at 0.989 and 0.99 for prediction in each of the two directions. This happens for reasons that are similar to those given above for 上巳 *zyoosi*. Hypothetical 刎呵 'decapitate' + 'scold' gets a low score because neither morpheme gets an opportunity to associate with the other during each of the single training instances that each morpheme participates in, the one for 呵 *ka* 'scold' being 啖呵 *tan-ka* 'defiant words'.

5 Testing hypothetical compounds

After training the model, I tested 1000 random combinations of morphemes that occur in the real compounds but do not occur together. The scores based on prediction of M_1 from M_2 ranged from 0.0518 at the low end to 0.7817 at the high end. Table 2 shows, in ascending order of scores, relevant data for the bottom 10 and top 10 among the 1000 hypothetical compounds tested.

Among the 10 lowest scoring compounds, only one shows up on a Google search: 6. 英瑚 *e-ko* is a possible girl's name found at NazukePON. The rest yield "No results found".

Among the 10 highest scoring compounds, 7 are found on the following Japanese web pages.[6]

992. 餅雪 *motu-yuki* is a pen name at Pixiv.

993. 廃話 is a song title at More Records.

994. 丁事 is at Open food facts[7]

996. 荒分*ara-bun* is personal name at Internet Archive.

997. 着書 is at Cultural Japan[8] with apparent meaning 'postal letter'.

998. 湯草 is at amazon.com as the name of a piece of pottery.

999. 仕品 is at The Japanese Association of Management Accounting with meaning 'project'.

**逐額 combines varied abstract meanings "pursue, drive away, chase, accomplish, attain, commit" + "forehead, tablet, plaque, framed picture, sum, amount, volume."

Given that we see a much higher incidence of hits in web searches among the high-scoring as opposed to low-scoring compounds, these limited results give a preliminary indication that the volume overlap ratio scores determined by the model tell how perspicuous the combination of two morphemes in a compound might be.[9]

M_1M_2	Score $M_2 \rightarrow M_1$	Freq M_1	Freq M_2	Glosses M_1, M_2
Shaded compounds were found in web pages.				
1. 玻倅	0.052	1	1	glass + son
2. 駱墳	0.065	1	1	white horse + tomb
3. 忌齦	0.077	5	1	mourning + grating teeth
4. 代濮	0.077	40	1	era + dredging
5. 茨褓	0.082	1	1	thorn + diaper
6. 英瑚	0.082	24	1	English + coral
7. 全堝	0.084	73	1	all + crucible
8. 裲吹	0.084	1	3	ancient robe + breathe
9. 堅躙	0.087	11	1	strict + trample
10. 英捕	0.089	24	2	English + capture
⋮				
991. 刑略	0.6695	7	26	punish + abbreviate
992. 餅雪	0.680	3	32	rice cake + snow
993. 廃話	0.687	39	47	abolish + speak
994. 丁事	0.695	12	100	* + thing
995. 制火	0.709	17	59	law + fire
996. 荒分	0.720	23	70	rough + part
997. 着書	0.730	36	83	wear + write
998. 湯草	0.736	28	43	hot water + grass
999. 仕品	0.745	11	41	serve + goods
1000. 逐額	0.782	6	26	**

Table 2: Bottom 10 and top 10 scores for model predictions of 1000 hypothetical compounds

[5] This hypothetical compound could be pronounced either with a Sino-Japanese pronunciation such as *sin-zi* or a native Yamato one as *kokoro-koto*.

[6] Some of these combinations of characters also show up on Chinese language webpages. These hits were not considered.

[7] Possible meanings for 丁 given at Nihongo Master are "street, ward, town, counter for guns, tools, leaves or cakes of someth[sic], even number, 4th calendar sign."

[8] This page was accessible on Chrome but throws an error when accessed via Firefox.

[9] A reviewer asks whether many of the high scoring compounds are "simply names", apparently questioning whether names are less constrained than other words in what morphemes can combine together. There is no obvious answer to this question, given that many names in the language suggest some interpretable meaning: e.g., *oo-saka* 'Osaka' 'big slope' or *kuro-sawa* 'Kurosawa' 'black swamp'. Conversely, many lexical compounds combine morphemes in ways that might seem implausible – e.g., *kei-setu* 螢雪 'firefly' + 'snow' = 'diligent study'.

Checking for internet search hits needs to be done manually by searching for the sought string of two characters on a page resulting from a Google search. One needs to sure of a number of things when searching: first, that the two characters are not, for example, occurring at the end and beginning of two consecutive phrases or sentences. One also needs to be sure that an M_1M_2 combination one is looking for is not occurring in an environment XM_1M_2Y, where XM_1 forms a compound and M_2Y forms a compound with an overall morphological structure $\{(XM_1)(M_2Y)\}$. In such a case M_1M_2 does not form a compound itself. And one also needs to be sure that the webpage one is checking is in Japanese and not Chinese, where the sought-after character sequence could also occur. Doing automated web-search results would provide us with much more data but it is questionable how accurate such data would be with respect to determining that a sought-after candidate compound actually occurs as a compound. As a result, I consider these web-search results as preliminary and show here only 10 examples from the bottom and top of the scale that were given a manual web search.

If we test these 20 examples for tetrachoric correlation between boolean variables 'yes/no for internet hit search' and 'occurs in top 10 vs. bottom 10 scores' we get a correlation result of 0.85. It should be noted that this data is underlyingly continuous: that is, not only are the score values continuous but the degree to which a hypothetical compound can be considered possible is also gradient, whether it is measured by number on internet hits or by native-speaker judgement scores.

Another problem with using internet search results as a test for the viability of a hypothetical compound is that whether or not such a compound is found does not necessarily determine how plausible it is. There could be some combinations that are not found that nevertheless would be judged possible by native speakers. On the other hand, some combinations that do occur are not necessarily forms that would enter general circulation in the language.[10] Accordingly, we can consider these results as preliminary evidence for the hypothesis that the model is learning, through box embeddings of morphemes, representations that can predict how well two morphemes can combine to form a compound word.

[10]This point was also noted by a reviewer.

5.1 Analysing the score results

I now investigate what kinds of associations between compound words and morphemes that the model finds in training might lead to high or low scores. If we take third-highest scoring hypothetical compound 湯草 *yu-kusa*[11] 'hot-water + grass' as an example, there are 56 real compounds for which the M_1 also forms a real compound with 草 *kusa* 'grass' and the M_2 also forms a real compound with 湯 *yu* 'hot water. An example is 青葉 *ao-ba* (green + leaf) 'fresh leaves' (also a placename) where 青 *ao* 'green' combines with 草 *kusa* in 青草 *ao-kusa* 'green grass' and 葉 *ha* 'leaf' combines with 湯 *yu* 'hot water' in 湯葉 *yu-ba*[12] 'tofu skin'.

青葉	*ao-ba*	real	green + leaf
青草	*ao-kusa*	real	green + grass
湯葉	*yu-ba*	real	hot water + leaf
湯草	*yu-kusa*	hypoth.	hot water + grass

This means that 湯 *yu* 'hot water' will tend to learn an embedding that is similar to the other M_1s that combine with a common set of M_2s. Similarly, M_2 草 *kusa* 'grass' will learn an embedding that is similar to the other M_2s that combine with this common set of M_1s. These sets of M_1 and M_2 embeddings will move closer to each other during training when many members of the two sets combine with each other in compounds, as is the case here. Clearly, having a large number of compounds in which each of 湯 *yu* 'hot water' and 草 *kusa* 'grass' occur increases the opportunity for this kind of association to occur, but frequency is not the only factor. For example, hypothetical compound 追竿 ('chase' + 'pole')[13] which has reasonably good morpheme frequencies of 39 and 6, got low scores from the model of only 0.154 and 0.237. If we search the corpus for other compounds in which the M_1 forms compounds with 竿 *sao* 'pole' and M_2 forms compounds with 追 *oi/tui* 'chase', we find no such compounds. 竿 *sao* 'pole' as an M_2 forms compounds in the corpus with morphemes 掛 *kake* 'hang', 竹 *take* 'bamboo', 鳥 *tori* 'bird', 秤 *hakari* 'balances', 旗 *hata* 'flag', 鰯 and *moti*

[11]This compound would be pronounced *on-soo* in a Sino-Japanese reading and *yu-kusa* in a native Yamato reading.

[12]The occurrence of an initial [b] on *ha* 'leaf' in the compound is a case of rendaku voicing, where /b/ is the voiced version of /h/.

[13]This compound is found on one single Japanese webpage at Excite Blog of haiku poems, where it appears to be more like a poetically licensed contraction of 'chased by a pole' than an actual compound.

'bird-lime' as M_1s but none of these forms a compound with any of the 39 M_2s that 追 *oi/tui* 'chase' combines with. This demonstrates that morpheme frequency is not the sole determiner of how well morphemes combine to make compounds. Also important are the associations between morphemes (or lack of them) that develop when morphemes occur together. The present model seeks to discover and encode those associations.[14][15]

6 Plotting overlap of morpheme embeddings

The plots in Figures 2 and 3 show graphically how the box embeddings of 18 of the above 20 pairs of morphemes overlap in the first 2 of 16 dimensions. Blue boxes are M_1s and red boxes M_2s.[16]

Because the plots show only the first 2 of 16 dimensions, the ratio of overlap volume to the volume of M_1 across all 16 dimensions will be lower than it appears in the plots. If the overlap ratio in each of 16 dimensions were 0.8, the overlap ratio of the total volume would be $0.8^{16} = 0.028$. Additionally, we don't see an exact progression in fraction of overlap as we proceed from the lowest to the highest scoring pair.[17]

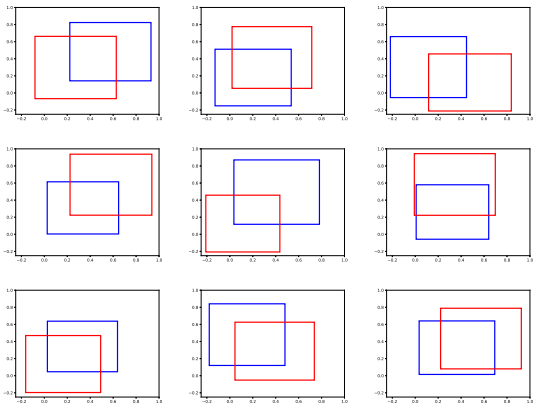


Figure 2: 9 lowest scoring hypothetical compounds

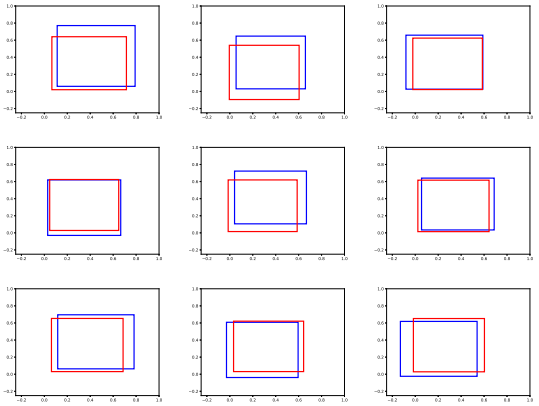


Figure 3: 9 highest scoring hypothetical compounds

7 Correlation with frequency?

Among the 10 lowest scoring and 10 highest scoring examples in Table 2, the lowest scorers all have one morpheme with a frequency of 1 and the highest scorers tend to have more frequently occurring morphemes. Can morpheme frequencies alone predict the viability of their combination in a compound? There are some combinations of relatively low frequency morphemes that score relatively high, one example being 蓮羽 *ren-ha* 'lotus + wing', which is the 31st highest scoring randomly composed compound in a list of 1000 random compounds, with frequencies of 6 and 7 for morphemes 蓮 *ren* 'lotus' and 羽 *ha* 'wing' . 蓮羽 *ren-ha* shows up on a Google search at Japanese Names Info as a possible boy's name. We also find the occasional low-scoring compound with at least one relatively high morpheme frequency: e.g., 追竿 'chase' + 'pole' in §5.1 above, scoring 0.154

[14] A reviewer suggests that an approach using collaborative filtering might be useful here. Arguably, the kinds of associations that develop in learning these embeddings would have a similar effect. As a further step, it would be useful to compare the present approach to one in which each morpheme is given a similarity score to another morpheme based on how many of the morphemes that each combines with in compounds are shared between the two. (See §8 for some initial steps in this direction.)

[15] A reviewer notes that there is an unlimited number of ways that the head and non-head in a two member compound could have a meaning relation and questions whether this model can "capture the range of possible meaning relations." It is not clear, though, that the precise meaning relation between M_1 and M_2 needs to be captured in order to predict whether such a compound could reasonably exist. What the model is learning is not necessarily the semantics of each morpheme but rather associations between morphemes that combine similarly with other morphemes. (See §8 below for further discussion.) For example, among the 56 above-mentioned compounds, the four whose M_1 is 下 'under' give the same adjectival meaning to 下, so the associations between similarly behaving morphemes seems to be more important there than precise meaning relations.

[16] One reviewer said that Figure 2 is "not that informative without knowing what the M_1 and M_2 scores are in each case." This comment misses the fact that scores only apply to the intersection of boxes of two morphemes and that morphemes themselves do not have scores in this model.

[17] Another reason that our calculations of volume will turn out to be a bit different from what is suggested by these graphs is that our code implements a SoftVolume function in the box embedding library of Chheda et al. (2021) that was developed by and discussed in Li et al. (2019) for dealing with the problem of getting disjoint boxes to overlap in training.

and 0.237 and frequencies of 39 and 6.

I tested scoring based on morpheme frequency with 1000 random out-of-corpus combinations of an M_1 and an M_2 and ordered them by the sum of the two morpheme frequencies. Results are shown in Table 3.

M_1M_2	$F_{M_1} + F_{M_2}$	Glosses
Shaded compounds were found in web pages.		
1. 嗜嬪	2	taste + bride
2. 誅違	2	death penalty + difference
3. 闖籟	2	inquire + sound of wind
4. 僭玄	2	arrogant + mysterious
5. 姻嚇	2	marriage + menacing
6. 齲芒	2	decayed tooth + pampas grass
7. 蛹蜂	2	chrysalis + cricket
8. 猩捺	2	orangutan + print
9. 菰咎	2	straw mat + blame
10. 廷忠	2	courts + loyalty
⋮		
991. 不闇	114	not + darkness
992. 秤色	118	scales + colour
993. 門金	118	gate + money
994. 不関	120	not + barrier
995. 公球	127	public + ball
996. 水谷	134	water + valley
997. 残日	135	remain + day
998. 孜子	146	industrious + child
999. 小諜	154	small + spy
1000. 閲人	168	inspection + person

Table 3: Bottom 10 and top 10 scores for model predictions of 1000 hypothetical compounds based on combined frequency of morphemes

None of the compounds in the lower-frequency section of the list was found on a web search. Among the 10 highest-scoring compounds, the following 4 were found in web searches:
992. at Agriknowledge meaning 'scale colour'.
994. at The Japanese Society of Chemotherapy with meaning 'indifferent'.
996. is a common surname.
997. at Nara Prefecture meaning 'remaining days'.
1000. as a personal name *etu-to* at Nazuke Pon

This is not a large statistical sample for comparing with the results from the box embedding model in Table 2 but four hits out of 10 in Table 3 is only marginally different from 7 in table 2, so further investigation is needed to determine whether morpheme frequency is as good a predictor as learned box embeddings.[18]

[18] A reviewer suggests that for further investigation, it might be better to count morpheme frequency based on how often a kanji character occurs in actual usage rather than in a lexicon of two-symbol words.

8 Exactly what is the model learning?

Following up on footnote 15, it is not clear that what the model is learning is at all the semantics of each morpheme. To test this, following Williams et al. (2020), and using the k-Nearest-Neighbour Information Estimator, (Gao, 2018), I tested the mutual information between the learned box embeddings for M_1s and each of (a) word2vec embeddings of the same morphemes, (b) the phonological information for each morpheme based on the hidden layer of a LSTM that was trained to predict its phonological string, (c) representations of the kanji characters as combinations of basic radical shapes taken from Breen (2020) and (d) a matrix of similarity scores between pairs of morphemes based on the number of M_2s that occur with both divided by the total number of M_2s that occur with ether of the two. The results suggest that semantics, to the extent it is encoded by word2vec embeddings, is not what the model is learning, with similarity scores and phonology showing the highest mutual information with the box embeddings.

Representations	MI
Word2vec with boxes	0.008
Phonology with boxes	0.129
Radicals with boxes	0.011
Similarity with boxes	0.229

Table 4: Mutual information calculations

9 Ideal pairings of M_2 with M_1

Table 5 below shows the top scoring 60 compounds in which an M_2 gets the highest volume-overlap ratio score with an M_1 in an out-of-corpus combination. Here, 45 out of 60, or 75% of the pairings are found in web searches. The last column shows the url of a page that contains the pairing of M_1 and M_2 if such a page was found. These results are not statistically conclusive but suggest that box embeddings that are learned on the basis of known morpheme combinations in compounds do contain information about morphemes that can predict how well they can combine to make a compound not seen in training.

M_1M_2	Score	Possible pronunciation	Gloss of each	Gloss of compound	Web link if found
1. 潰脹	0.9268	*kai-tyoo*	crush + dilate	pen name	Pixiv
2. 東岸	0.9075	*too-gan*	east + coast	'east coast'	Nihongo Master
3. 旺脹	0.9049	*oo-sin/kyoo-sin*	flourishing + flourishing		
4. 地風	0.9044	*ti-kaze*	earth + wind	personal name	Nazuke Pon
5. 現用	0.9009	*gen-yo*	currently + used	'currently used'	NihongoMaster
6. 部主	0.9006	*bu-syu*	part + master		Buddhism
7. 国人	0.8976	*koku-zin*	country + person	'indigenous person'	Japandict
8. 人風	0.8975	*zin-huu/nin-huu*	person + wind	a pen name	Tik Tok
9. 空山	0.8972	*sora-yama*	sky + mountain	a family name	Pinterest
10. 重作	0.895	*zyuu-saku*	heavy + work	'heavy work'	Your katakana
11. 手面	0.8924	*te-zura*	hand + surface	name	Japanese Names Info
12. 別国	0.8909	*betu-koku*	other + country	'another country'	Asian Historical Records
13. 下面	0.8875	*ka-men*	under + surface	'underside'	Romaji Desu
14. 三風	0.8873	*san-puu*	three + wind	store name in Koriyama	Yelp
15. 自学	0.887	*zi-gaku*	self + study	'self-study'	JapanDict
16. 上幅	0.8865	*zyoo-huku*	upper + width	'upper width'	BigLemon
17. 全作	0.8852	*zen-saku*	all + work	'whole work'	JLearn
18. 難意	0.8846	*nan-i*	impossible + thought		
19. 多調	0.8845	*ta-tyoo*	many + tone	'polytonal'	JapanDict
20. 一作	0.8841	*is-saku*	one + make	family name	Worldcat
21. 愧瑣	0.8837	*oo-sa*	distress + small, chain		
22. 毎春	0.8811	*mai-haru*	every + spring	'every spring'	Weblio
23. 有学	0.88	*u-gaku*	exist + study	Buddhist term	Japanese Wiki Corpus
24. 当位	0.8795	*too-i*	correct + rank		
25. 内家	0.8791	*nai-ka*	inside + house		
26. 外学	0.8787	*gai-gaku*	outside + study		
27. 出部	0.8777	*de-bu*	leave + part	family name	Your Katakana
28. 美種	0.8768	*yosi-tane*	beautiful + seed	name	Pon Navi
29. 心体	0.8752	*sin-tai*	heart + body	a name of a performance	Taka Takiguchi
30. 回戦	0.8747	*kai-sen*	times + battle	'match, game'	Romaji Desu
31. 学社	0.8736	*gaku-sya*	study + company		Pinterest
32. 家家	0.8733	*ie-ie*	house + house	'every house'	Romaji Desu
33. 軍費	0.8731	*gun-pi*	war + expenditures	'war funds'	Japandict
34. 中面	0.8728	*naka-tura*	middle + surface	family name	National Cancer Centre
35. 本所	0.8725	*hon-syo*	main + office	'main office'	JapanDict
36. 神利	0.8721	*kami-ri*	divine + profit	family name	Fate Grand Order Wiki
37. 各産	0.8721	*kaku-san*	each + product	'each product'	LP Gas
38. 二端	0.8716	*ni-tan*	two + edge		
39. 仏名	0.8712	*butu-myoo*	Buddha + name	'Buddha's name'	JapanDict
40. 用利	0.8702	*yoo-ri*	use + profit		
41. 大心	0.87	*tai-sin*	big + heart	boy's name	Japanese Names Info
42. 同利	0.8697	*doo-ri*	same + profit		
43. 通意	0.8696	*tuu-i*	pass through + idea	'meaning'	Cultural Japan
44. 無調	0.8689	*mu-tyoo*	not + tone	'atonality'	Japan Wikipedia
45. 良道	0.8683	*yosi-miti*	good + path	name	Nazuke Pon
46. 遠座	0.8678	*en-za*	distant + seat	family name	Japanese Names Info
47. 小風	0.8678	*ko-huu*	small + wind	'breeze'	Tanoshii Japanese
48. 雑論	0.8668	*gen-ron*	miscellany + discussion	'miscellaneous remarks'	Genron blog
49. 産部	0.8666	*san-bu*	product + part		
50. 経科	0.8664	*kee-ka*	sutra + department		
51. 議略	0.8663	*gi-ryaku*	opinion + abbreviation		
52. 土屋	0.8658	*tuti-ya*	earth + door	family name	Lingq
53. 西辺	0.8644	*nisi-be*	west + sides	family name	Japanese Names
54. 金屋	0.8638	*kana-ya*	metal, money + room	place name found in numerous locations	
55. 定産	0.8633	*tee-san*	fixed + production	'regular production'	Issu
56. 高食	0.8632	*koo-syoku*	high + food		
57. 主話	0.863	*syu-wa*	master + speak	'main discourse'	Spotify
58. 総訳	0.8623	*soo-yaku*	full + translation	'general translation'	CiNii
59. 楽書	0.8623	*raku-gaki*	easy + write	'graffiti'	JapanDict
60. 会学	0.8622	*kai-gaku*	meet + study		

Table 5: Top 60 scores for model predictions of the best scoring hypothetical compound for each M_1 in the corpus

10 Comparison with a vector embedding model

For comparison, I ran the same data through a model that used simple vector embeddings rather than box embeddings. Since each dimension of the 16D box embeddings consists of two values: one for each of z_i and Z_i, (the maximum and minimum coordinates of the box in each dimension), to make the comparison fair I used an embedding dimension of 32 for vector embeddings. The model was trained on 9 million data points in the same way as with the box embeddings. The score of a combination of two morphemes was the sigmoided dot product of their two embedding vectors. The objective was to bring the score for a real compound close to 1.0 and for a non-occurring one to 0.0.

Testing the learned embeddings on a random sample of 1000 out-of-dataset compounds as was done for trained box embeddings, we find that among the top 40 scorers, only 4 yield web search hits, and out of the top 10, only the last one (牛流 *go-ryuu*) is found in a web search. This result compares unfavourably with the web-search results for trained box embeddings in Table 2.

If we look at the constituent meanings among top 10 scorers just mentioned (Table 6), the semantic juxtapositions appear no more odd than the pairs of meanings in the bottom half of Table 2 that had hypothetical compounds scored on box embeddings. The lack of any perceptible difference in semantic congruity between the top scorers in the two models supports the conclusion that what the model is learning is not so much semantics but rather associations between morphemes based on co-occurrence in known compounds as discussed above in §5.1. A possible clue for why box embeddings might better encode these associations than simple vectors is that, as mentioned above on page 2, they can capture negative correlations between concepts (in this case between morphemes that tend not to occur together) through non-overlap of boxes in a way not possible with simple vectors (Vilnis et al., 2018).

Compd.	Possible pronunciation	M_1	M_2
駐部	*tyoo-bu*	reside	part
端成	*tan-see*	edge	become
芸山	*gee-san*	art	mountain
変本	*hen-bon*	strange	origin
轆行	*rok-koo*	pulley	go
海語	*kai-go*	ocean	language
称木	*syoo-moku*	praise	tree
伏作	*huku-saku*	prostrate	make
国車	*koku-sya*	country	wheel
牛流	*go-ryuu*	cattle	method

Table 6: Constituent morphemes in top 10 scoring hypothetical compounds trained on vector embeddings

11 Next steps

The initial steps for training a model of box embeddings of morphemes that occur in compound words are offered here as a proof-of-concept to build on in further research. Given the limitations of evaluating the model with webpage hits, a next step is to elicit native-speaker judgements of hypothetical compounds that are evaluated by the model. To what extent might such judgement scores correlate with those given by the model? I would also like to experiment with different hyperparameters of the model. Does increasing the dimension size of the box embeddings enable the model to better capture relations between morphemes or does the enlarged space make it too difficult to get boxes to overlap where we wish them to?

Since box lattice embeddings were first proposed by Vilnis et al. (2018), they have been mainly used for tasks like hypernym prediction, for example, in Li et al. (2019). To my knowledge, the present study is the first instance of their implementation for the task of encoding abstract properties of morphemes based on which other morphemes they associate with in compound word formation. This study shows the promise of opening up new possibilities for how box embeddings might encode a speaker's knowledge of language.

Acknowledgements

Thanks to members of Paul Smolensky's Neurosymbolic Computation Lab Group and three anonymous reviewers for helpful comments and feedback. Research was generously funded by IGRA grant at U. Leipzig. All errors are my own.

References

Jim Breen. 2020. RADKFILE/KRADFILE. "This project provides a decomposition of kanji into a number of visual elements or radicals to support software which provides a lookup service using kanji components."'.

Tejas Chheda, Purujit Goyal, Trang Tran, Dhruvesh Patel, Michael Boratko, Shib Sankar Dasgupta, and Andrew McCallum. 2021. Box embeddings: An open-source library for representation learning using geometric structures.

Weihao Gao. 2018. knnie (k-Nearest Neighbor Information Estimator).

Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. 2019. Smoothing the geometry of probabilistic box embeddings. In *International Conference on Learning Representations*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Akiko Nagano and Masaharu Shimada. 2014. Morphological theory and orthography: Kanji as a representation of lexemes. *Journal of Linguistics*, 50(2):323–364.

Andrew Nathaniel Nelson. 1987. *The Modern Reader's Japanese English Character Dictionary*. Charles E. Tuttle Company.

NHK. 2016. *The NHK Japanese Pronunciation Dictionary*. Japanese Broadcasting Corporation (NHK).

Dhruvesh Patel, Shib Sankar Dasgupta, Michael Boratko, Xiang Li, Luke Vilnis, and Andrew McCallum. 2020. Representing joint hierarchies with box embeddings. In *Automated Knowledge Base Construction*.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 263–272. Association for Computational Linguistics.

Adina Williams, Tiago Pimentel, Hagen Blix, Arya D. McCarthy, Eleanor Chodroff, and Ryan Cotterell. 2020. Predicting declension class from form and meaning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6682–6695, Online. Association for Computational Linguistics.

Neural Networks Can Learn Patterns of Island-insensitivity in Norwegian

Anastasia Kobzeva
Norwegian University of Science and Technology
anastasia.kobzeva@ntnu.no

Suhas Arehalli
Johns Hopkins University
suhas@jhu.edu

Tal Linzen
New York University
linzen@nyu.edu

Dave Kush
University of Toronto
dave.kush@utoronto.ca

Abstract

Recent research suggests that Recurrent Neural Networks (RNNs) can capture abstract generalizations about filler-gap dependencies (FGDs) in English and so-called *island* constraints on their distribution (Wilcox et al., 2018, 2021). These results have been interpreted as evidence that it is possible, in principle, to induce complex syntactic knowledge from the input without domain-specific learning biases. However, the English results alone do not establish that island constraints were induced from distributional properties of the training data instead of simply reflecting architectural limitations independent of the input to the models. We address this concern by investigating whether such models can learn the distribution of acceptable FGDs in Norwegian, a language that is sensitive to fewer islands than English (Christensen, 1982). Results from five experiments show that Long Short-Term Memory (LSTM) RNNs can (i) learn that Norwegian FGD formation is unbounded, (ii) recover the island status of temporal adjunct and subject islands, and (iii) learn that Norwegian, unlike English, permits FGDs into two types of embedded questions. The fact that LSTM RNNs can learn cross-linguistic differences in island facts therefore strengthens the claim that RNN language models can induce the constraints from patterns in the input.

1 Introduction

Human linguistic knowledge is complex and abstract, yet children master language relatively easily and quickly through exposure to their native language(s). A major debate centers around whether acquiring such knowledge requires complex domain-specific learning biases or whether it can be induced from the input using domain-general learning routines. We contribute to this debate by investigating whether Recurrent Neural Networks (RNNs), which are weakly biased language models, can induce complex knowledge of filler-gap dependencies and constraints on them from the input in Norwegian.

Filler-Gap Dependencies (FGDs) are contingencies between a displaced filler phrase and a later gap position where the filler is interpreted (denoted with __ throughout the paper). There are different types of FGDs. (1-a) is a *wh*-FGD where the filler *wh*-word is interpreted as the direct object of the verb *forged*. (1-b) is a Relative Clause (RC) FGD where the filler, the head of the RC, *painting*, is interpreted as the direct object of *forged* within the RC.

(1) a. They found out what the dealer forged __ using a new technique.
b. They found the painting that the dealer forged __ using a new technique.

FGDs have been the subject of extensive research because they require complex hierarchical generalizations about sentence structure to be interpreted. For example, establishing the RC FGD in (1-b) requires (i) identifying the head of the RC as a filler corresponding to a later empty NP position; (ii) knowing that *forged* requires a direct object; (iii) identifying the gap by recognizing the absence of an object next to *forged*, and (iv) associating the filler with the gap to form a dependency. There is a bidirectional relationship between the filler and the gap: fillers require gaps to be interpreted, and gaps require fillers to be properly licensed. This relationship can be established across a potentially unbounded structural distance as in (2).

(2) She knows what he thought they found out the dealer forged __ using a new technique.

FGDs are also constrained. Certain environments, called *islands* (Ross, 1967), block FGD formation. Various structures have been identified

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 175-185.
Amherst, Massachusetts, June 15-17, 2023

as islands. For example, embedded questions (3-a), sentential subjects (3-b), and adjuncts (3-c) are generally considered island domains in English.

(3) a. *What did he wonder [whether the dealer forged __]?
b. *What is [that the dealer forged __] extremely likely?
c. *What does the dealer worry [if they find out __]?

How do learners acquire island constraints? Nativist approaches hold that acquisition of islands would be impossible without innate domain-specific learning biases due to the induction problem known as the Poverty of the Stimulus (PoS; e.g., Chomsky 1986; Crain and Pietroski 2001). According to this argument, the input to the learner lacks direct evidence that islands exist. The input is therefore compatible with conflicting hypotheses about whether islands should be in the adult target state. The fact that learners nevertheless converge on the same set of island constraints has led the proponents of the nativist approach to suggest that innate domain-specific learning biases guide learners to the conclusion (for example, Subjacency Condition, Chomsky 1973).

Empiricist approaches, on the other hand, claim that the input is sufficiently rich to support learning island constraints when coupled with domain-general learning biases (Clark and Lappin, 2010). This position has recently gained support from neural network simulations. Wilcox and colleagues suggest that RNNs (and other autoregressive neural models) can capture the abstract generalizations governing *wh*-FGDs in English, as well as the associated island constraints (2018; 2019b; 2019a; 2021). They claim that this result militates against the PoS argument that islands cannot be induced from the input without domain-specific biases.

Wilcox and colleagues' results are suggestive, but they do not fully establish that the models 'learn' islands from the input. An alternate explanation is that the results are artifacts. Under this possibility, RNNs do not pursue FGDs into islands in English because the models are simply incapable of representing syntactic dependencies into island environments irrespective of the input they receive (either because the domains are too complex or because of some other unknown limitation inherent to the RNN architecture). One way of ruling out this explanation is to test the models' performance on a language that has a different set of island constraints. If the models can learn to pursue FGDs in another language into domains that are islands in English, that would constitute additional evidence that the models are inducing islands from the input.

To this end, we explore whether RNNs can learn the distribution of acceptable FGDs and island constraints in Norwegian – a language that differs from English in the set of domains that are islands. To preview our results, the models can learn that temporal adjuncts and subject phrases are islands in Norwegian, but that embedded questions are not (*wh*-islands). These results suggest that weakly-biased RNNs can capture patterns of island-insensitivity in Norwegian, thus providing empirical evidence that this pattern of cross-linguistic variation can be learned from the input.

2 Island constraints in Norwegian

Norwegian is similar to English in several respects when it comes to FGDs. Norwegian allows long-distance dependencies with gaps in various syntactic positions. Norwegian also exhibits sensitivity to some of the same islands that English does. FGDs into temporal adjuncts (4) or subject phrases (5) are unacceptable in Norwegian like English (Bondevik et al., 2021; Kush et al., 2019, 2018; Kobzeva et al., 2022b).

(4) *Hva spiste du kake [da han spiste __]?
What ate you cake when he ate __
*'What did you eat cake when he ate __?'

(5) *Hva har [brevet om __] skapt problemer?
What has letter.DEF about __ created problems
*'What has the letter about __ created problems?'

On the other hand, Norwegian allows FGDs into environments that are considered islands in English, such as Embedded Questions (EQs, Christensen 1982; Maling and Zaenen 1982). RC FGDs into embedded constituent questions like (6) are found in written corpora of Norwegian (Kush et al., 2021) and native speakers rate various types of FGD into EQs as acceptable in judgment studies (Kobzeva et al., 2022b).

(6) Vi var redde for noe vi ikke visste [hva __ var].
We were afraid of smth we NEG knew what __ was.
'We were afraid of something we did not know what __ was.'

This distribution of FGDs in Norwegian makes it a good testing ground for exploring whether RNNs can induce a set of islands that is different from what is observed in English. Recent research shows that RNNs can capture basic generalizations about *wh*- and RC FGDs in Norwegian: they learn that fillers can license gaps in different syntactic

positions and across increased linear distance between the filler and the gap (Kobzeva et al., 2022a). Here we expand on this line of research by testing whether RNNs can learn that FGDs like (6) are acceptable in Norwegian, while simultaneously ruling out FGDs like (4) and (5). We do so by testing whether the models are less likely to expect FGDs in potential island environments relative to control sentences without island structures. We also test the robustness of the result by testing two more models with the same architecture but different initializations.

We ran five experiments. Experiment 1 tested whether the models learn that Norwegian FGDs are unbounded by seeing if they can successfully associate fillers and gaps across multiple embedded clauses. Establishing this basic result is a prerequisite for testing islands, which typically require cross-clausal dependencies. Experiments 2 and 3 tested if the models can learn that temporal adjunct clauses and complex subject phrases are islands in Norwegian, as in English. Finally, Experiments 4 and 5 tested if RNNs can learn that FGDs into embedded questions are possible in Norwegian. Experiments 1-4 evaluate the models performance on Norwegian only, while Experiment 5 directly compares *wh*-FGDs in Norwegian and English.

3 Method

3.1 Language models

We trained Long Short-Term Memory (LSTM) RNNs (Hochreiter and Schmidhuber, 1997) to take a sequence of words as input and compute a probability distribution of the next word over the model's vocabulary. We trained three such models with different random initializations following the procedure described in (Gulordava et al., 2018), using the code provided by the authors[1]. Each model was a 2-layer LSTM with 650 hidden units in each layer, trained for 40 epochs on 113 million tokens of Norwegian Wikipedia (in the Bokmål written standard) with a vocabulary size of 50000 most frequent words. The models achieved perplexities between 30.05 and 30.3 on the validation set.

3.2 Dependent measure

We test how the models would fare as incremental language processors by looking at *surprisal*, which measures how (un)predictable a word is given a specific prompt using the models' probability distribution. We measure the surprisal values by computing the negative log of the predicted conditional probability from the models' softmax layer.

3.3 Measuring FGDs

Wilcox et al. (2018) introduced a 2×2 factorial design for measuring FGDs inspired by psycholinguistic paradigms. The design independently manipulates the presence of a filler and the presence of a gap as in (7).

(7) They found out...
- a. that the dealer forged the art -FILLER, -GAP
- b. *what the dealer forged the art +FILLER, -GAP
- c. *that the dealer forged __ -FILLER, +GAP
- d. what the dealer forged __ +FILLER, +GAP

...using a new technique.

When both the filler and the gap are absent (7-a) or present (7-d), the sentences are grammatical. When either the filler or the gap is absent, (7-b) and (7-c), the sentences are ungrammatical. We measure *filler effects* – how the presence of a filler affects surprisal – in two different pairwise comparisons. *Filled gap effects* are measured by comparing surprisal associated with an NP in -GAP conditions. *Unlicensed gap effects* are measured by comparing surprisal associated with a gap in the +GAP conditions. We discuss each type of filler effect in more detail below.

3.3.1 Filled gap effects

In behavioral studies, filled gap effects are regarded as support for the *active gap-filling* strategy: after encountering a filler, the processor actively predicts a gap without waiting for the actual gap site. Stowe (1986) observed a slow-down in self-paced reading times at the direct object *us* in (8-b), which contains the filler *who*, compared to the same word in a corresponding sentence without a filler (8-a). The slow-down reflects a violated expectation: seeing a filler caused the processor to predict a gap in object position.

(8) a. My brother wanted to know if Ruth will bring *us* home to Mom at Christmas.
b. My brother wanted to know who Ruth will bring *us* home to __ at Christmas.

We test whether the models exhibit similar filled gap effects. We measure the surprisal difference between the ungrammatical +FILLER, -GAP condition as in (7-b) and the grammatical -FILLER, -GAP condition in (7-a) at the region of the filled NP (*the*

[1]https://github.com/facebookresearch/colorlessgreenRNNs

art in (7)). If seeing a filler sets up an expectation for a gap in object position, the NP should be more surprising in (7-b) than in (7-a), resulting in a *positive* surprisal difference.

Crucially, humans do not exhibit filled gap effects inside island environments (Stowe, 1986; Traxler and Pickering, 1996; Phillips, 2006), indicating that the active prediction of gaps is suspended where they are impossible. Following the same logic, if the models show sensitivity to island constraints, we expect to see no filled gap effects inside islands.

3.3.2 Unlicensed gap effects

Unlicensed gap effects provide a measure of how 'surprised' the model is to encounter a gap without a filler to license it. We measure these effects as a difference in surprisal between the grammatical +FILLER, +GAP (7-d) condition and ungrammatical -FILLER, +GAP (7-c) condition at the region following the gap (*using a new technique* in (7)). If a presence of a gap without a licensing filler is surprising to the models, the unlicensed gap effect should manifest as a negative difference between low surprisal in the post-gap region in (7-d) and high surprisal in (7-c).

Unlicensed gap effects show if the models recognize gaps as licit inside certain syntactic environments. Whereas filled gap effects measure the models' expectation for an upcoming gap, unlicensed gap effects arguably should reflect the models' understanding of grammaticality, as sentences with illicit gaps are ungrammatical (and, unlike filled gaps, cannot be 'rescued' by establishing another gap site later in a sentence). Analogous to filled gap effects, unlicensed gap effects should be close to zero in island environments if the models can derive their island status from their training data.

3.4 Statistical analysis

Following standard practice in psycholinguistics, statistical analysis was performed using mixed-effect linear regression models with sum-coded fixed effects of FILLER (0.5 for +FILLER, -0.5 for -FILLER) and CONDITION (0.5 for CONTROL and -0.5 for ISLAND except for Experiments 1 and 4, see details below). We fit the statistical models on differences in surprisal between +FILLER, -FILLER conditions with these fixed effects and a maximal random effect structure (Barr et al., 2013). We ran separate models for filled gap effects in the filled NP region and for unlicensed gap effects in the post-gap region. If a model failed to converge, we reduced the random effect structure until convergence was reached. Model formulas are presented in Appendix A.

4 Experiments

4.1 Experiment 1: Unboundedness

It is important to establish whether LSTMs can represent FGDs across hierarchical distance before testing island environments, as they involve cross-clausal dependencies. Therefore, in Experiment 1 we tested how increased hierarchical distance between the filler and the gap influences models' representations of FGDs. To do that, we manipulated the number of clausal embeddings between the filler and the gap (from 1 to 5 layers of clausal embedding, as illustrated in (9)). We created 30 items by crossing the factors FILLER and GAP in (7) with NUMBER OF LAYERS, resulting in a 2 × 2 × 5 design. Test sets were created for *wh*- and RC FGDs (600 test sentences per dependency type).

(9) a. 1 LAYER (+FILLER, +GAP)
Hun vet hva selgeren forfalsket __ ved hjelp
She knows what dealer.DEF forged __ with help
av moderne teknologi.
of modern technology.
'She knows what the dealer forged __ using modern technology'.

b. 5 LAYERS (+FILLER, +GAP)
Hun vet hva han trodde de fant ut
She knows what he thought they found out
avisen rapporterte politiet visste
newspaper.DEF reported police.DEF knew
selgeren forfalsket __ ved hjelp av moderne
dealer.DEF forged __ with help of modern
teknologi.
technology.
'She knows what he thought they found out the newspaper reported the police knew the dealer forged __ using modern technology'.

We tested all three models on all of the items, and we present the results averaged across the models for both dependency types together. Overall, filler effects decrease as layers of embedding increase (Figure 1). For *wh*-dependencies (blue bars), there was a significant reduction in both the filled gap effect and the unlicensed gap effect already at two layers of embedding, which was also true for every layer thereafter (p's <0.05 in all cases). For RC dependencies (orange bars), there was a significant reduction in filled gap effects at three layers (p <0.05), and in unlicensed gap effects at two layers (p's <0.001) of sentential embedding, as well as for every layer thereafter (p's <0.001 in all cases).

Tables with statistics summary can be found in Appendix A.

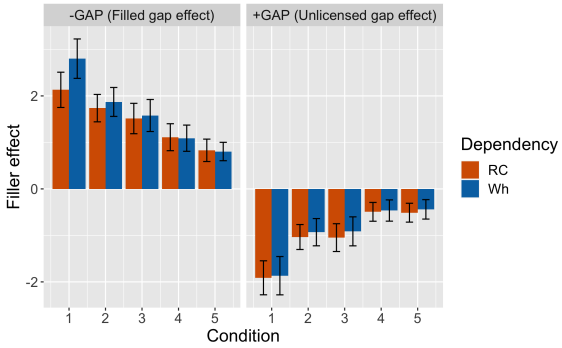


Figure 1: Unboundedness experiment: Filler effects by the number of embeddings for both dependency types. Bars represent an average over three models, error bars represent 95% confidence intervals.

Despite the reduction in filler effects as a function of the number of sentential embeddings, the filler effects remain above zero even at the largest hierarchical distance. This suggests that the models have learned that FGD formation is unbounded and have the basic representational capacity required for testing FGDs inside islands.

4.2 Islands shared between Norwegian and English

Experiments 2 and 3 tested FGDs into constituents that are islands in Norwegian (just as in English) – subjects and temporal adjunct clauses – to see if the models' expectations for FGDs are attenuated within the two environments in Norwegian, as previously seen in English (Wilcox et al., 2018, 2021).

4.2.1 Experiment 2: Subject island

Fillers cannot be associated with gaps inside a subject phrase, like the gap inside the prepositional phrase attached to the subject in (10). Such sentences are rated as unacceptable by English speakers, and the same pattern is found in Norwegian (11-b). We compare the island condition in (11-b) to an NP-subject extraction as in (11-a).

(10) *The newspaper reported what [the agreement with __] will strengthen the political interaction after the elections.

(11) a. SUBJECT CONTROL (+FILLER, +GAP)
Avisen rapporterte hva som __ vil
Newspaper.DEF reported what REL __ will
forsterke det politiske samspillet etter
strengthen the political interaction.DEF after
valget.
election.DEF
'The newspaper reported what __ will strengthen the political interaction after the election.'

b. SUBJECT ISLAND (+FILLER, +GAP)
*Avisen rapporterte hva [avtalen
Newspaper.DEF reported what agreement.DEF
med __] vil forsterke det politiske
with __ will strengthen the political
samspillet etter valget.
interaction.DEF after election.DEF
'*The newspaper reported what the agreement with __ will strengthen the political interaction after the election.'

We created 30 items according to a $2 \times 2 \times 2$ design that crossed the factors FILLER and GAP in (7) with a third factor: CONDITION (CONTROL, ISLAND). Again we created separate sets of sentences for *wh*- and RC FGDs (240 total test sentences per dependency type). The results of this experiment are presented in Figure 2.

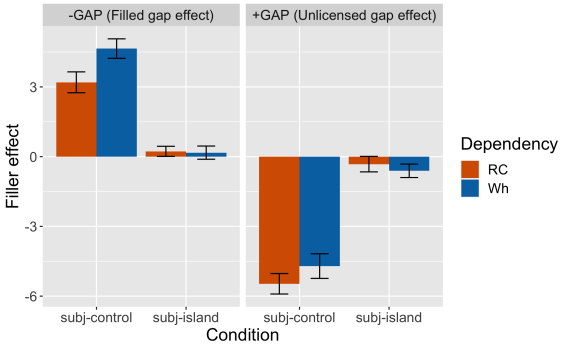


Figure 2: Subject island experiment: Filler effects by gap position for both dependency types.

Filled gap effects (Figure 2 left panel) were large in the control condition, but were significantly reduced in the island condition: statistical analysis revealed a main effect of CONDITION for both dependency types (both p's <0.001). The same pattern was found for unlicensed gap effects (Figure 2 right panel). For both dependency types, there was a significant effect of CONDITION (p's <0.001 in both cases). These results show that the models exhibit reduced filler effects within subject islands, which is in line with behavioral acceptability data from native Norwegian speakers.

4.2.2 Experiment 3: Adjunct island

Adjuncts are said to block FGD formation, which explains the unacceptability of (12): The filler *what* cannot be associated with the gap inside the adjunct *when*-clause. Norwegian, like English, does not al-

low gaps inside temporal adjuncts (Bondevik et al., 2021; Bondevik and Lohndal, 2023).

(12) *What were the voters excited [when the politician visited __ last week]?

We created 30 items according to a 2 × 2 × 3 design that crossed FILLER, GAP, and CONDITION for each dependency type (360 test sentences per dependency). CONDITION had three levels that determined the location of a direct object gap. In the LINEAR CONTROL (13-a) and STRUCTURAL CONTROL (13-b) the gap was not embedded in an island, whereas in ADJUNCT ISLAND (13-c), the gap was embedded inside a temporal adjunct (headed by *mens 'while', da 'when', etter at 'after'* and *før 'before'*). In the linear control condition (13-a), first used in (Wilcox et al., 2018), the filler and gap are in the same clause, but the linear distance between them is comparable to the distance in (13-c). In the structural control condition (13-b), our novel addition to the design, the filler and the gap are separated across two clauses, making the *structural* distance between the filler and the gap comparable to (13-c). We included these control conditions in order to estimate the independent effects of linear distance and structural distance on the model's performance, so as to better isolate island effects.

(13) a. LINEAR CONTROL (+FILLER, +GAP)
Jeg husker hva politikeren med godt
I remember what politician.DEF with good
omdømme besøkte __ forrige uke.
reputation visited __ last week.
'I remember what the politician with a good reputation visited __ last week.'

b. STRUCTURAL CONTROL (+FILLER, +GAP)
Jeg husker hva avisen rapporterte at
I remember what newspaper.DEF reported that
politikeren besøkte __ forrige uke.
politician.DEF visited __ last week.
'I remember what the newspaper reported that the politician visited __ last week.'

c. ADJUNCT ISLAND (+FILLER, +GAP)
*Jeg husker hva velgerne var begeistret
I remember what voters.DEF were excited
da politikeren besøkte __ forrige uke.
when politician.DEF visited __ last week.
'*I remember what the voters were excited when the politician visited __ last week.'

We defined two contrasts for analysis: CONTROL contrast compared effect size between the two control conditions (linear vs. structural). ISLAND contrast compared effects between the structural control and the adjunct island condition.

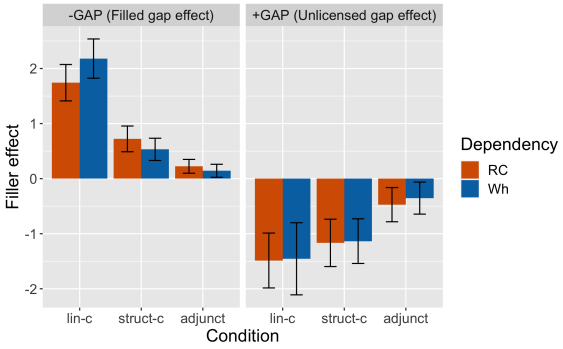


Figure 3: Adjunct island experiment: Filler effects by condition for both dependency types. Control conditions are `lin-c` and `struct-c`.

The results of the experiment are presented in Figure 3. Filled gap effects for both dependency types (left panel) were largest in the linear control condition, significantly larger than in the structural control condition (CONTROL contrast p's <0.001). Filled gap effects were in turn significantly larger in the structural control condition than in the adjunct island condition (ISLAND contrast p's <0.001), where filled gap effects were close to zero.

The same qualitative pattern was observed with unlicensed gap effects for both dependency types (right panel). Unlicensed gap effects were larger in the linear control condition compared to the structural control, and in the structural control condition compared to the island condition (p's <0.001 in all cases). Therefore, the models show reduced filler effects inside temporal adjuncts in Norwegian. However, the average filler effects are not 0 in the adjunct island condition, suggesting that the models might not treat them as full islands.[2] Norwegian shows some variation in adjunct island effects, with extraction from conditional adjuncts rated higher than from temporal and reason-adjuncts (Bondevik et al., 2021; Bondevik and Lohndal, 2023). The result obtained here could be explained by the models' sensitivity to this variation (and potential overgeneralization).

4.3 Islands contrasting English and Norwegian

The results of Experiments 2 and 3 suggest that the models learn that subjects and temporal adjuncts are islands in Norwegian, similar to the conclusions

[2]On around 65% of the trials, the models show filled-gap effects greater than zero, while unlicensed gap effects are less than zero on around 70% of the trials. However, the effects are mostly small, under 1 bit of surprisal 90% of the time.

made for English by Wilcox et al.. Experiments 4 and 5 test whether the models can learn that embedded questions (EQs) are not islands in Norwegian. We test two types of EQs in Norwegian: 1) interrogative EQs, and 2) *whether*-EQs.

4.3.1 Experiment 4: Interrogative EQ

According to Kush et al. (2021), the most common type of extraction from EQs (in a children's fiction corpus) includes a subject gap inside an interrogative EQ as in (14).

(14) Vi var redde for noe vi ikke visste [hva __ var].
We were afraid of smth we NEG knew what __ was.
'We were afraid of something we did not know what __ was.'

We chose to first test such EQs because we reasoned that they were likely the most frequent in the model's training data. We created 30 items that crossed FILLER, GAP, and CONDITION for each dependency type (240 test sentences per dependency). CONDITION controlled whether the embedded clause was an EQ (15-b) or a declarative complement (15-a) control.[3]

(15) a. DECLARATIVE CONTROL (+FILLER, +GAP)
Han sa hvem som sjåføren glemte at __
He said who REL driver.DEF forgot that __
skulle hentes i sentrum den dagen.
should be.picked.up in center.DEF that day.DEF.
'He said who$_i$ the driver forgot (that) __$_i$ should be picked up in the center that day.'

b. WH-ISLAND (+FILLER, +GAP)
Han sa hvem som sjåføren glemte hvor __
He said who REL driver.DEF forgot where __
skulle hentes __ den dagen.
should be.picked.up that day.DEF.
'He said who$_i$ the driver forgot where$_k$ __$_i$ should be picked up __$_k$ that day.'

We expected clear filled gap effects and unlicensed gap effects in the declarative clauses. If the models recognize that interrogative EQs are not islands in Norwegian, the filled gap effects and unlicensed gap effects in the EQ sentences should be comparable to their declarative counterparts, or at least greater than zero.

[3]The direct translation of (15-b) would be ungrammatical in English due to *that*-trace effects. Norwegian exhibits some variation in *that*-trace effects; theoretical and experimental work shows that it mostly allows subject gaps after *that* (Lohndal, 2009; Kush and Dahl, 2020). We return to this issue in the Discussion.

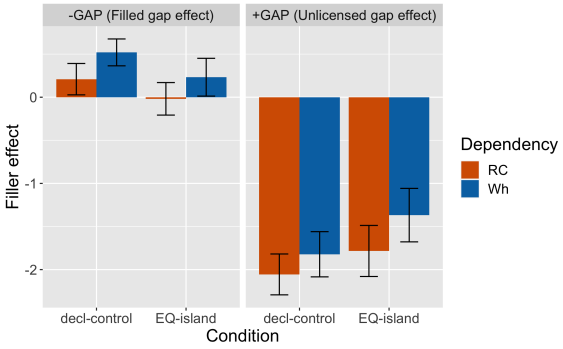


Figure 4: Interrogative EQ island experiment: Filler effects by condition for both dependency types.

Figure 4 shows that filled gap effects were small or close to 0 across all conditions and dependency types, while unlicensed gap effects were large. Statistical analysis revealed a main effect of CONDITION for both filled gap effects and unlicensed gap effects with *wh*-dependencies (p's <0.01). With RC dependencies, the same was true for the filled gap effect (p <0.05, orange bars on the left panel). For the unlicensed gap effect with RC dependencies, the effect of CONDITION was not significant (p <0.1). Importantly, despite the significant effect of CONDITION in three out of four cases tested, both filled gap effects and unlicensed gap effects in the island condition were comparable to the declarative control, suggesting that the models treat EQs and embedded declarative clauses similarly with respect to FGD formation in Norwegian.

4.3.2 Experiment 5: *Whether*-EQ

In Experiment 4, we tested FGDs into interrogative EQs with gaps in subject position. However, previous research in English has not tested interrogative EQs and has instead focused on FGDs into polar EQs, *whether*-islands. For example, Wilcox et al. tested *whether*-islands with gaps in object position in English. An example of +FILLER, +GAP, ISLAND condition from their *whether*-island experiment is presented in (16).

(16) *I know what my brother said whether our aunt devoured __ at the party.

In order to facilitate more direct cross-linguistic comparison, and to test the robustness of the result of Experiment 4, we decided to run an experiment comparing FGDs into *whether*-EQs in English and Norwegian side by side. To do so, we slightly modified the 24 English items from (Wilcox et al., 2018) and created 24 novel items following the same tem-

plate, resulting in 48 items total. We then translated them into Norwegian. As the original (Wilcox et al., 2018) items did not include RC dependencies, we restricted dependency types to *wh*-FGDs in this experiment. We compared the performance of the Gulordava model (used by Wilcox et al., 2018) on English stimuli and the performance of one of the Norwegian models (used by Kobzeva et al., 2022a). The results are presented in Figure 5.

Overall, filler effects are smaller in English (light blue bars) than in Norwegian (dark blue bars; main effect of LANGUAGE, p <0.001). The pattern of island sensitivity also differs. In Norwegian, robust filled gap effects were observed in both declarative control and *whether*-island environments, while in English, no filled gap effect was observed inside a *whether*-island (left panel). Statistical analysis confirmed a significant CONDITION × LANGUAGE interaction for filled gap effects (p <0.01). Similar differences were observed for unlicensed gap effects (right panel): In Norwegian, unlicensed gap effects are equally large in declarative complements and *whether*-islands, whereas there is no unlicensed gap effect inside a *whether*-island in English compared to the declarative control (CONDITION × LANGUAGE p <0.05).

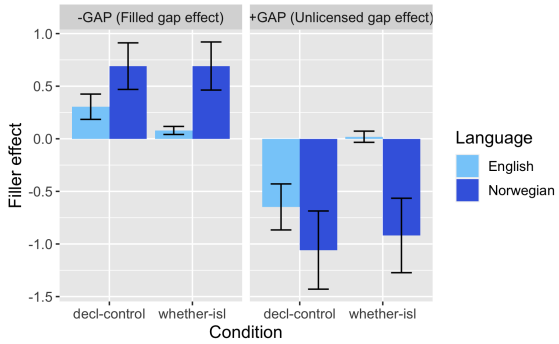


Figure 5: *Whether*-island experiment (with *wh*-dependencies): Comparison of filler effects in English and Norwegian.

Taken together with the fact that the architecture of the English and the Norwegian model was the same, and that they were trained using the same hyper-parameter combination for the same number of epochs on input data that were comparable in size and genre, these results suggest that RNNs can come to different conclusions about the status of *whether*-islands based on different language input. This provides further evidence for the claim, made in Wilcox et al., that autoregressive language models can learn the distribution of FGDs in a language from their input.

5 Discussion

In this paper, we tested LSTMs' ability to establish FGDs in Norwegian by looking at filled gap effects and unlicensed gap effects. Experiment 1 found non-zero filled gap effects and unlicensed gap effects across multiple layers of embedding suggesting that the models learn that FGDs are unbounded. Experiments 2 and 3 showed that filled gap effects and unlicensed gap effects are significantly reduced inside subject phrases and temporal adjuncts, suggesting that the models learned that these domains are islands in Norwegian, mirroring previous findings for English (Wilcox et al., 2018, 2019a,b, 2021).

Broadly speaking, results from Experiments 4 and 5 suggest that the models can learn that embedded questions are not island environments in Norwegian. In both Experiment 4 and 5, we found large unlicensed gap effects in Norwegian interrogative EQs and in Experiment 5 we observed filled gap effects inside Norwegian *whether*-EQs. Taken together, the results are consistent with the conclusion that LSTM RNNs can learn cross-linguistic differences in island facts from different language input. We do not know whether the model's generalization was derived from actual examples of FGDs into embedded questions in the training data, or whether the model learned the distribution indirectly. We cannot verify that in this case that the models learned from direct evidence, but it is plausible that such evidence would be available in the Wikipedia corpus given that FGDs into embedded questions are found (in relatively small numbers) in other corpora (such as the child fiction corpus investigated by Kush et al., 2021).

One potentially surprising finding was the asymmetry in filled and unlicensed gap effects between Experiments 4 and 5. In Experiment 4, filled gap effects were not robust in subject position, but unlicensed gap effects were. In Experiment 5, both filled gap effects and unlicensed gap effects were observed in object position. We take this effect to mean that the model was not actively pursuing embedded subject gaps in our stimuli. There are various possible interpretations for this effect. One possibility is that the model avoids gaps after overt material in left edge of a clause (a kind of *that-trace* effect, see Lohndal, 2009). Another

possibility is that embedded subject gaps were not frequent enough in the training data to establish strong expectations for them.

We do not take the fact that filled gap effects are absent in some EQs as evidence against the models being able to establish FGDs into EQs. Even in the absence of filled gap effects, unlicensed gap effects show that the models can still recognize gaps in EQs as licit in Norwegian. We think that unlicensed gap effects provide a better indication of what the models have learned is possible. In other words, the two effects measure different aspects related to an FGD: While filled gap effects measure active expectation/prediction for a gap inside a particular structural configuration (i.e. whether the models think that a gap is *likely* in a given position), unlicensed gap effects reflect whether the models 'understand' that FGDs are in principle possible in that configuration. We suggest that future work using this paradigm should keep this dissociation in mind when interpreting results: Learning what a possible FGD is, does not necessarily entail active expectation in RNN language models.

One outstanding question is how well the model's active gap-filling behavior mirrors how actual humans would process these sentences. Native English speakers do not actively pursue gaps inside islands (Stowe, 1986; Traxler and Pickering, 1996; Phillips, 2006). In this regard, the English models of Wilcox et al. mimic human behavior. It is unknown whether native Norwegian speakers suspend active gap-filling inside islands, but pursue active gap-filling inside structures like EQs, that are not islands in their language. Future work should test the alignment between the model's performance and human behavior.

6 Conclusion

In this study, we tested whether LSTMs, an RNN architecture without language-specific bias, can learn two types of filler-gap dependencies in Norwegian in several (potential) island environments. We found evidence that the models can pick up patterns of island-insensitivity when it comes to embedded questions in Norwegian, while still inducing island effects in subject and adjunct islands. Our results also show that RNNs are sensitive to differences in the distribution of FGDs in English and Norwegian, suggesting that the input to the models must provide enough evidence for the diverging patterns. Our results lead us to reassess the importance of domain-specific learning biases in acquiring island constraints from the input.

Acknowledgements

All experimental materials and analysis scripts are available in the project's OSF repository. Our models were trained with the resources from NTNU's IDUN computing cluster (Själander et al., 2019). We thank the members of NTNU's EyeLands Lab (Øyelab) for their comments on an earlier version of this paper. We also thank Sigurd Farstad Iversen for his help in creating the stimuli in Norwegian.

References

Dale J Barr, Roger Levy, Christoph Scheepers, and Harry J Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3):255–278.

Ingrid Bondevik, Dave Kush, and Terje Lohndal. 2021. Variation in adjunct islands: The case of Norwegian. *Nordic Journal of Linguistics*, 44(3):223–254.

Ingrid Bondevik and Terje Lohndal. 2023. Extraction from finite adjunct clauses: an investigation of relative clause dependencies in norwegian. *Glossa: a journal of general linguistics*, 8(1).

Noam Chomsky. 1973. Conditions on transformations. In Morris Halle, Stephen R. Anderson, and Paul Kiparsky, editors, *A Festschrift for Morris Halle*, pages 232–286. Holt, Rinehart and Winston, New York.

Noam Chomsky. 1986. *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group.

Kirsti Koch Christensen. 1982. On multiple filler-gap constructions in Norwegian. In Elisabet Engdahl and Eva Ejerhed, editors, *Readings on unbounded dependencies in Scandinavian languages*, pages 77–98. Almquist & Wiksell, Stockholm.

Alexander Clark and Shalom Lappin. 2010. *Linguistic Nativism and the Poverty of the Stimulus*. John Wiley & Sons.

Stephen Crain and Paul Pietroski. 2001. Nature, nurture and universal grammar. *Linguistics and philosophy*, 24(2):139–186.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of NAACL 2018*, pages 1195–1205.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Anastasia Kobzeva, Suhas Arehalli, Tal Linzen, and Dave Kush. 2022a. LSTMs Can Learn Basic Wh- and Relative Clause Dependencies in Norwegian. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Anastasia Kobzeva, Charlotte Sant, Parker T. Robbins, Myrte Vos, Terje Lohndal, and Dave Kush. 2022b. Comparing island effects for different dependency types in Norwegian. *Languages*, 7(3):195–220.

Dave Kush and Anne Dahl. 2020. L2 transfer of L1 island-insensitivity: The case of Norwegian. *Second Language Research*, pages 1–32.

Dave Kush, Terje Lohndal, and Jon Sprouse. 2018. Investigating variation in island effects: A case study of Norwegian wh-extraction. *Natural Language & Linguistic Theory*, 36(3):743–779.

Dave Kush, Terje Lohndal, and Jon Sprouse. 2019. On the island sensitivity of topicalization in Norwegian: An experimental investigation. *Language*, 95(3):393–420.

Dave Kush, Charlotte Sant, and Sunniva Briså Strætkvern. 2021. Learning island-insensitivity from the input: A corpus analysis of child- and youth-directed text in Norwegian. *Glossa: a journal of general linguistics*, 6(1):1–50.

Terje Lohndal. 2009. Comp-t effects: Variation in the position and features of C. *Studia Linguistica*, 63(2):204–232.

Joan Maling and Annie Zaenen. 1982. A phrase structure account of Scandinavian extraction phenomena. In Pauline Jacobson and Geoffrey K. Pullum, editors, *The Nature of Syntactic Representation*, pages 229–282. Springer Netherlands, Dordrecht.

Colin Phillips. 2006. The real-time status of island phenomena. *Language*, pages 795–823.

John Robert Ross. 1967. *Constraints on variables in syntax*. PhD dissertation, MIT.

Magnus Själander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. 2019. EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure.

Laurie A Stowe. 1986. Parsing wh-constructions: Evidence for on-line gap location. *Language and cognitive processes*, 1(3):227–245.

Matthew J Traxler and Martin J Pickering. 1996. Plausibility and the processing of unbounded dependencies: An eye-tracking study. *Journal of Memory and Language*, 35(3):454–475.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019a. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP*, pages 181–190.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019b. What syntactic structures block dependencies in RNN language models? *arXiv preprint arXiv:1905.10431*.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN Language Models Learn about Filler-Gap Dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*, pages 211–221.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2021. Using computational models to test syntactic learnability. *Linguistic Inquiry*, pages 1–88.

A Results of Statistical Tests

The levels of significance used in the tables below: + $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. The statistics are presented separately for filled gap effects (FGE) and unlicensed gap effects (UGE) by each dependency type and experiment. The response variable *s* in `lmer` formulas is the difference in surprisal between +FILLER, -FILLER conditions.

1. Unboundedness

s ~lyrs + (1+lyrs | model) + (1+lyrs | item)

	FGE, *wh*-dependencies		
	Est.	S.E.	t
(Intercept)	2.801	0.304	9.221***
layers2	−0.931	0.220	−4.240*
layers3	−1.223	0.204	−5.980***
layers4	−1.711	0.246	−6.959***
layers5	−1.997	0.219	−9.104***
	UGE, *wh*-dependencies		
(Intercept)	−1.867	0.147	−12.681***
layers2	0.936	0.099	9.488***
layers3	0.954	0.099	9.671***
layers4	1.402	0.099	14.212***
layers5	1.427	0.099	14.465***
	FGE, RC dependencies		
(Intercept)	2.131	0.194	10.971***
layers2	−0.394	0.281	−1.402
layers3	−0.617	0.237	−2.598*
layers4	−1.019	0.203	−5.024***
layers5	−1.301	0.233	−5.593**
	UGE, RC dependencies		
(Intercept)	−1.912	0.192	−9.954***
layers2	0.877	0.161	5.447***
layers3	0.864	0.156	5.557***
layers4	1.419	0.166	8.564***
layers5	1.400	0.158	8.885***

2. Subject island			
s ~cond + (1+cond | model) + (1+cond | item)			
	FGE, *wh*-dependencies		
	Est.	S.E.	t
(Intercept)	2.411	0.255	9.459***
condition	4.476	0.335	13.368***
	UGE, *wh*-dependencies		
(Intercept)	−2.658	0.255	−10.437***
condition	−4.098	0.488	−8.390***
	FGE, RC dependencies		
(Intercept)	1.713	0.132	12.944***
condition	2.970	0.254	11.697***
	UGE, RC dependencies		
(Intercept)	−2.895	0.223	−13.008***
condition	−5.147	0.383	−13.455***

3. Adjunct island			
s ~cntrs + (1+cntrs | model) + (1+cntrs | item)			
	FGE, *wh*-dependencies		
	Est.	S.E.	t
(Intercept)	0.952	0.127	7.476***
controlCntrs	2.457	0.232	10.609***
islandCntrs	1.618	0.221	7.323***
	UGE, *wh*-dependencies		
(Intercept)	−0.981	0.208	−4.714***
controlCntrst	−0.948	0.298	−3.182**
islandCntrst	−1.255	0.273	−4.602***
	FGE, RC dependencies		
(Intercept)	0.896	0.136	6.593***
controlCntrst	1.692	0.200	8.454***
islandCntrst	1.344	0.234	5.755***
	UGE, RC dependencies		
(Intercept)	−1.042	0.187	−5.569***
controlCntrst	−0.889	0.201	−4.423***
islandCntrst	−1.139	0.178	−6.382***

4. Interrogative EQ			
s ~cond + (1+cond | model) + (1+cond | item)			
	FGE, *wh*-dependencies		
	Est.	S.E.	t
(Intercept)	0.376	0.081	4.647***
condition	0.288	0.107	2.690**
	UGE, *wh*-dependencies		
(Intercept)	−1.595	0.260	−6.142***
condition	−0.454	0.153	−2.961**
	FGE, RC dependencies		
(Intercept)	0.095	0.080	1.189
condition	0.228	0.100	2.271*
	UGE, RC dependencies		
(Intercept)	−1.920	0.220	−8.707***
condition	−0.272	0.152	−1.795+

5. *Whether*-EQ			
*s ~condition*language + (1+condition | item)*			
	FGE		
	Est.	S.E.	t
(Intercept)	0.617	0.074	8.388***
condition	0.109	0.102	1.074
language	0.700	0.102	6.880***
condition:language	−0.625	0.204	−3.073**
	UGE		
(Intercept)	−0.652	0.099	−6.570***
condition	−0.354	0.132	−2.690**
language	−0.676	0.127	−5.346***
condition:language	0.627	0.253	2.477*

Noise-tolerant learning as selection among deterministic grammatical hypotheses

Laurel Perkins
Department of Linguistics
University of California Los Angeles
Los Angeles, CA 90095-1543
perkinsl@ucla.edu

Tim Hunter
Department of Linguistics
University of California Los Angeles
Los Angeles, CA 90095-1543
timhunter@ucla.edu

Abstract

Children acquire their language's canonical word order from data that contains a messy mixture of canonical and non-canonical clause types. We model this as noise-tolerant learning of grammars that deterministically produce a single word order. In simulations on English and French, our model successfully separates signal from the noise introduced by non-canonical clause types, in order to identify that both languages are SVO. No such preference for the target word order emerges from a comparison model which operates with a fully-gradient hypothesis space and an explicit numerical regularization bias. This provides an alternative general mechanism for regularization in various learning domains, whereby tendencies to regularize emerge from a learner's expectation that the data are a noisy realization of a deterministic underlying system.

1 Introduction

Children at early stages of language acquisition draw accurate grammatical generalizations from incomplete, immature, and variable representations of their input. For example, infants learn their language's basic word order despite immature abilities to identify clause arguments, and despite non-canonical constructions that disrupt this basic word order (e.g., *wh*-questions, passives) (Hirsh-Pasek and Golinkoff, 1996; Perkins and Lidz, 2020, 2021). This is one of many ways in which learners draw generalizations that are more regular or deterministic than the variable data that they are learning from. What kind of mechanisms allow for learning to abstract away from messiness in (the learner's representation of) the data?

One potential answer emerges from studies of learning in the context of unpredictable variability, for example in the context of acquiring language from non-native speakers. This approach posits a general learning bias to *regularize* inconsistent variability (Hudson Kam and Newport, 2005, 2009; Reali and Griffiths, 2009; Culbertson et al., 2013; Ferdinand et al., 2019). Learners consider hypotheses that closely match the statistical distributions in their input, but in some circumstances they are biased to "sharpen" those distributions, pushing them towards more systematic extremes.

Implicit in this account is a hypothesis space that can accommodate the full variability of the data. For instance, when exposed to an artificial language in which determiners occur inconsistently with nouns, children are equipped to consider that the language allows determiners with any probability, but nonetheless prefer to use particular determiners all of the time or not at all (Hudson Kam and Newport, 2005, 2009). The literature takes this as evidence for a regularization bias operating within a learner's fully-flexible hypothesis space, pushing learners to prefer probabilities closer to 0 or 1 and producing near-categorical learning outcomes. This idea could be applied to the learning of basic word order in infancy— for example, learning that English is canonically SVO. Children who encounter a messy mixture of canonical and non-canonical sentences would be equipped to consider that clause arguments can flexibly occur in multiple orders in the language, but prefer hypotheses that are skewed heavily towards one consistent order.

Here, we explore a different approach. We propose that in certain circumstances, learners face a choice among discrete hypotheses, each of which is deterministic in a way that is incompatible with the full variability of the observed data. Learners expect that their data result from an opaque interaction between (i) one of the deterministic hypotheses currently under consideration, and (ii) various other processes that might introduce "noise" into the data. For a child learning an artificial determiner system, the data might reflect a combination

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 186-198.
Amherst, Massachusetts, June 15-17, 2023

of signal for deterministic rules, and noise coming from unknown grammatical or extra-grammatical processes. For a child learning the syntax of basic clauses, the data reflects a combination of signal for the target language's basic word order and noise introduced by non-canonical sentence types. Regularization emerges when learners are able to successfully identify signal for a deterministic hypothesis within their noisy data (Perkins et al., 2022; Schneider et al., 2020).

We introduce a general computational framework for performing this inference. A learner of the sort we describe below expects that its data are generated by a complex system: a core deterministic component that the learner is attempting to acquire, operating alongside a "noise" component whose properties are currently unknown. Using the case study of basic word order acquisition, we show that our model can learn to separate evidence for a deterministic grammar of canonical word order from the distorting effects of non-canonical noise processes. It does so without knowing ahead of time how much noise there is, or what its properties are. Moreover, we show that our approach fares better in this learning problem than the more common approach to regularization described above. This suggests that in certain domains, successful learning from noisy data is enabled by a hypothesis space comprising restrictive grammatical options.

2 The intuition behind our approach

Suppose that a bag contains coins of two types: Type A coins, which always come up heads, and Type B coins, which all have some single unknown probability θ of coming up heads. We know nothing about how many of each type are in the bag. We observe ten coin flips, producing eight heads and two tails. How many of these flips might we guess came from Type A coins, and how many from Type B coins? There is a wide range of options, including the possibility that all ten flips came from Type B coins; but given the observed skew towards heads, there is a clear intuition that Type A coins were probably responsible for a significant portion of the observations. Why is this?

Under the hypothesis that all ten flips came from Type B coins, eight of those flips would need to come up heads and two to come up tails in order to generate the observed data. Contrast this with the (more intuitively plausible) hypothesis that there were six Type A and four Type B flips. Under this hypothesis, the six Type A flips need to come up heads, which is guaranteed to happen; so, generating the observed data just amounts to having the four Type B flips produce two heads and two tails. This is clearly less "costly" than the first hypothesis's requirement that ten Type B flips produce eight heads and two tails. By positing six Type A flips, six of the heads that we need to generate "come for free"; with only Type B flips, however, we get no such head start.

More precisely, the *likelihood* of the observed data, under the hypothesis that relies on only four Type B flips, is $\binom{4}{2}\theta^2(1-\theta)^2$. Under the hypothesis that leaves all the work to ten Type B flips, this likelihood is $\binom{10}{8}\theta^8(1-\theta)^2$. It is the exponents that matter: the ten-flip likelihood is smaller than the four-flip likelihood whenever $\theta < 0.71$, so for most values of θ. This is one way to understand our intuitive preference for hypotheses that invoke Type A flips. We can make this even more precise by *marginalizing* over θ; see Appendix A for details. These details make clear that *all* that matters about a particular hypothesis is how many Type B flips it must appeal to. We've seen that four Type B flips is better than ten, but two is even better: the very best hypothesis is that there were eight Type A flips and two Type B flips (likelihood $(1-\theta)^2$).

Suppose now that, as well as the bag with two-headed coins and head-tail coins (call this Bag H), there is a bag with two-*tailed* coins and head-tail coins (Bag T). We again see 10 coin flips, 8 heads and 2 tails. We know that they all came from one of the two bags, and we have to guess which one.

We have seen that Bag H makes available "good" explanations of the data, which exploit the presence of two-headed coins to minimize the crucial number of uncertain head-tail flips. With Bag T, however, the available "known outcome" coins produce tails; so the best we can do is to suppose that both of the two observed tails came from the two-tailed coins, and rely on eight uncertain flips to do the rest of the work (likelihood θ^8). Since there is no way for the two-tailed coins to contribute to a good explanation of the observed high proportion of heads, Bag H is a better guess than Bag T.

This choice between Bag H and Bag T will correspond to the choice between competing restricted hypotheses in the learners we describe below. It will be useful to think of this as essentially a choice between the two-headed coin and the two-tailed coin, where either choice (since it's accompanied

by head-tail coins) is embedded in a system that also produces some "noise", i.e. divergences from what would be generated by these core mechanisms alone. When comparing such composite systems, our learner will prefer the one whose core mechanisms predict the skew in the data; this will provide the least costly solution, even though the shared noise possibilities ensure that all the competing systems can account for the data as a whole. And the proposed learner will do this without knowing *a priori* how much of the data is noise (i.e. how much of the data came from the head-tail coins) or what the contribution of noise looks like (i.e. the probability θ of noise contributing a head).

Perkins et al. (2022) applied this approach to model how learners might identify the core transitivity properties of verbs in their language, despite "noise" from non-canonical clause types. This type of noise might arise when a young child encounters an obligatorily-transitive verb in a sentence with a displaced object (e.g., *What did you bring?*) but is unable to parse it as such. By hypothesizing that unknown noise processes cause the data to be a distorted reflection of verbs' core argument-taking properties, their model was able to successfully identify that certain verbs were deterministically transitive and intransitive— for roughly the same reason that Bag H above provides a good explanation for data that does not consist entirely of heads.

Here, the basic syntax we consider generates subjects and objects according to some canonical order (SVO, SOV, etc.), yielding surface strings of verbs and noun phrases. And just like in Perkins et al., unknown grammatical processes— for instance, argument movement or ellipsis— operate alongside this basic syntax, with the result that the observed strings of verbs and noun phrases are a distorted reflection of canonical word order.

3 Applying this to PCFGs

We now turn to situations where a learner's core hypotheses take the form of grammars — specifically, probabilistic context-free grammars (PCFGs). The learner will observe some collection of strings, and in general none of the core grammars under consideration will be consistent with all of the observed strings. One way to apply the idea from above would be to suppose that some of the observed strings were generated by a separate "noise grammar" — just as some of the coin flips above were generated by the head-tail coin. But this would mean that every string is analyzed as either all signal (i.e. informative about the core grammar) or all noise, and so the learner would not be able to extract useful information from subparts of sentences.

Instead, we allow the signal-or-noise choice to be made at a finer-grained level: each derivational step might be contributed either by the core hypothesized grammar or by noise processes. Either way, each step is licensed by a CFG-style rewrite rule; in other words, the noise is itself characterized by particular rules for expanding nonterminals that sit alongside the rules of the core grammar. The overall system therefore consists of rules of two sorts, which we'll call *core rules* and *noise rules*.

Framed slightly more generally: we formulate a generative process for strings that we call a *Mixture PCFG*. A Mixture PCFG uses rules built out of terminal and nonterminal symbols in the manner of a standard PCFG. But whereas defining a standard PCFG involves identifying just a single set of rules, defining a Mixture PCFG involves identifying two sets of rules. For the moment we will simply call them ϕ-rules and ψ-rules, but in the case study below these will correspond to core rules and noise rules respectively. A particular candidate rewriting step, e.g., 'S $\rightarrow$ NP VP', might be included in the set of ϕ-rules, in which case it will have some non-zero probability $\phi_{\text{S}\rightarrow\text{NP VP}}$ associated with it; and independently might be included in the set of ψ-rules, in which case it will have some non-zero probability $\psi_{\text{S}\dashrightarrow\text{NP VP}}$ associated with it. In addition to these ϕ parameters and ψ parameters, a Mixture PCFG has one additional parameter ϵ^A associated with each nonterminal symbol A, which controls the choice between using a ϕ-rule or a ψ-rule to expand an occurrence of A.

To illustrate, an example Mixture PCFG is shown in Fig. 1. In this grammar and all those in the case studies below, NP is deterministically realized as np and V as v; we abstract away from these steps in all the discussion that follows.[1] We write ϕ-rules with standard arrows and ψ-rules with dashed arrows. Notice that the ϕ-probabilities associated with the expansions of a particular nonterminal symbol sum to one, as do the ψ-probabilities. Roughly foreshadowing the grammars we use in the case study below, the ϕ-rules in Fig. 1 encode the basic clause structure of an SVO language, and the ψ-rules generate "noise" that diverges from this canonical word order in various ways.

[1] This is just $\epsilon^{\text{NP}} = \epsilon^{\text{VP}} = 0$ and $\phi_{\text{NP}\rightarrow\text{np}} = \phi_{\text{V}\rightarrow\text{v}} = 1$.

ϕ-rules	ψ-rules	ϵ probabilities
1.0 S $\rightarrow$ NP VP	0.3 S $\dashrightarrow$ VP NP	$\epsilon^{\text{S}} = 0.2$
	0.5 S $\dashrightarrow$ NP S	
	0.2 S $\dashrightarrow$ VP	
0.4 VP $\rightarrow$ V	0.7 VP $\dashrightarrow$ NP V	$\epsilon^{\text{VP}} = 0.3$
0.6 VP $\rightarrow$ V NP	0.3 VP $\dashrightarrow$ NP	

Figure 1: An example Mixture PCFG.

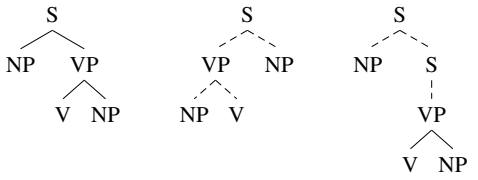


Figure 2: The three possible analyses of np v np (supressing NP$\rightarrow$np and V$\rightarrow$v rewrites).

To calculate the probability of a string under this Mixture PCFG, we sum over all possible ways it can be generated. For the string np v np, for example, there are three possibilities, shown in Fig. 2; solid lines represent expansions using ϕ-rules, and dashed lines expansions using ψ-rules.

The first tree represents one way of generating np v np that uses only ϕ-rules: ϵ^{S} is the probability of using a ψ-rule rather than a ϕ-rule to expand an occurrence of S, and so the probability of expanding the root S node as shown in this first tree is the product of $(1 - \epsilon^{\text{S}})$ and the corresponding ϕ-probability. The probability of the entire tree is the product of two such rewrites, as in (1); similarly, the probability of the second tree is given in (2). The third tree's probability, in (3), uses a more interesting combination of ϕ-rules and ψ-rules.

$$(1)\quad (1-\epsilon^{\text{S}})(\phi_{\text{S}\rightarrow\text{NP VP}}) \times (1-\epsilon^{\text{VP}})(\phi_{\text{VP}\rightarrow\text{V NP}})$$

$$(2)\quad (\epsilon^{\text{S}})(\psi_{\text{S}\dashrightarrow\text{VP NP}}) \times (\epsilon^{\text{VP}})(\psi_{\text{VP}\dashrightarrow\text{NP V}})$$

$$(3)\quad (\epsilon^{\text{S}})(\psi_{\text{S}\dashrightarrow\text{NP S}}) \times (\epsilon^{\text{S}})(\psi_{\text{S}\dashrightarrow\text{VP}}) \times (1-\epsilon^{\text{VP}})(\phi_{\text{VP}\rightarrow\text{V NP}})$$

Using values from Fig. 1, these three trees therefore have probabilities of 0.336, 0.013 and 0.001, respectively; and so the total probability of the string np v np is 0.350.[2]

Although we are restricting attention to PCFGs here, exactly the same approach could be used to formulate "mixture" versions of any kind of probabilistic grammar where the probability of a complex structure is the product of the probabilities of certain local choices (e.g. HMMs or PFSAs). The sampling methods we employ below for inference are compatible with any model where these local choices are expressed as multinomial distributions.

In the learning scenarios modeled below, the learner will have some set of hypotheses to choose from, each of which is represented by a Mixture PCFG such as that in Fig. 1. One of the competitor hypotheses might be represented by a similar Mixture PCFG that has the basic clause structure of an SOV language (rather than SVO) reflected in its ϕ-rules, and has some of the same ψ-rules as Fig. 1. Each of these two hypotheses will therefore generate strings that diverge from the strict SVO or SOV pattern licensed by its particular ϕ-rules. Deciding which of these two Mixture PCFGs provides a better explanation of some observed strings is therefore analogous to the decision between Bag H and Bag T in Section 2, with the ϕ-rules corresponding to the two-headed and two-tailed coins, and the ψ-rules corresponding to the head-tail coins.[3] Just as the decision between Bag H and Bag T could be made by considering (i.e. marginalizing over) all possible values of the unknown weight θ, we can make the decision between competing Mixture PCFGs in a way that considers all possible ϕ, ψ and ϵ values. The logic outlined in Section 2, whereby explanations in terms of core mechanisms that align with skews in the data are preferred, carries over to the case where the core mechanisms are either SVO or SOV word order.

4 Case study: Learning basic word order

We show that the approach of deciding among competing Mixture PCFGs provides a novel solution to the problem of word order acquisition in early development. Children acquire the basic word order of their language from data that contains a large amount of noise. For example, English learners identify that their language is canonically SVO in infancy, before they can identify the processes that produce non-canonical word orders in sentences like *wh*-questions (Hirsh-Pasek and Golinkoff, 1996; Lidz et al., 2017; Perkins and Lidz, 2020, 2021). Many accounts assume that learners have the ability to "filter" non-basic sentences of this sort, ignoring them when drawing

[2]The overall mechanics of a Mixture PCFG can be recast as a single classical PCFG. Specifically: add nonterminals S_ϕ and S_ψ alongside S, and include the rules $\text{S} \rightarrow \text{S}_\phi$ and and $\text{S} \rightarrow \text{S}_\psi$ with probabilities $(1 - \epsilon^{\text{S}})$ and ϵ^{S}, respectively; the subsequent expansions of S_ϕ and S_ψ are determined by the ϕ-rules for S and the ψ-rules for S, respectively. Our implementation in fact works with exactly this classical PCFG.

[3]Bag H is analogous to a Mixture PCFG with $\phi_{S\rightarrow\text{h}} = 1$, $\psi_{S\dashrightarrow\text{h}} = \theta$ and $\psi_{S\dashrightarrow\text{t}} = 1 - \theta$, and ϵ^{S} representing the proportion of head-tail coins in the bag.

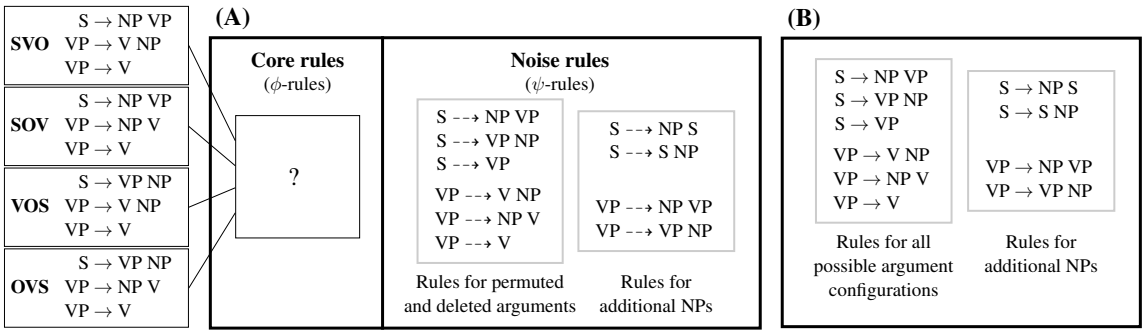


Figure 3: (A) Hypothesis space for our noise-tolerant learner; (B) Fully-flexible learner for comparison.

early syntactic inferences (e.g. Pinker, 1984). But if learners do not yet know what counts as basic, how do they identify which sentence types count as *non*-basic, in order to filter them out (Gleitman, 1990; Perkins et al., 2022)? Our model provides a way to implement the essence of this filtering idea, while avoiding potential issues of circularity.

Our learner's hypothesis space consists of four sets of ϕ-rules and one shared set of ψ-rules, giving rise to the four Mixture PCFGs in Fig. 3A. The ϕ-rules generate the core predicate-argument structure of basic transitive and intransitive clauses, deterministically putting subjects before or after verb phrases and objects before or after verbs. This yields a 4-way choice of canonical word order: SVO, SOV, VOS, OVS.[4] Subjects are are obligatory and objects are optional, reflecting the learner's belief that canonical clauses need subjects. All four grammars share the same set of noise rules, which allow for all permutations and deletions of NP arguments, and for additions of NPs into non-argument positions. The flexibility in the noise rules produces many more possibilities for expanding a given nonterminal than are provided by the core rules, mirroring the asymmetry between restrictive two-headed (and two-tailed) coins and flexible head-tail coins.

Crucially, while the learner's noise rules contain hypotheses about which non-canonical processes might operate in its language, the learner does not know ahead of time the ψ and ϵ probabilities associated with these rules: it does not know which kinds of non-canonical clauses it will encounter, or how frequently. We show that our learner is able to identify the correct Mixture PCFG— the correct combination of core and noise rules— using only the distributions of noun phrases and verbs that a 15-month-old infant might be able to represent. This inference does not require information about underlying clause structure. However, a similar mechanism could be generalized to make use of structural cues from meaning or prosody (Pinker, 1984; Christophe et al., 2008).

Using strings of imperfectly-identified noun phrases and verbs, the learner evaluates the following three questions, corresponding to the ϕ, ψ, and ϵ parameters of its input filter, respectively: (1) What do the data from the core rules look like? (2) What do the data from the noise rules look like? (3) What is the right division into signal vs. noise? For each grammar in its hypothesis space, the learner considers the possible answers to these questions in order to determine how well that grammar explains the data it observes. Comparing across the four grammars, the learner selects the grammar that provides the best explanation.

4.1 Generative model

The model's data consists of a collection $\vec{w}$ of strings, each comprising a single v with any number of satellite np's (i.e., of the form np^* v np^*). The model assumes that these are generated by one of the Mixture PCFGs in its hypothesis space (Fig. 3A), each of which has equal prior probability; the learner is not biased *a priori* in favor of any particular word orders.

Given any particular Mixture PCFG, we can construct an equivalent standard PCFG that defines the same distribution over strings (via some additional nonterminals and unary rules; see Footnote 2 for details). Let $\vec{\theta}^{A_G}$ be the vector of weights of the allowable expansions of a given nonterminal A in this resulting standard PCFG G; the prior over $\vec{\theta}^{A_G}$

[4]We limit our focus to these four word orders because they are the options generated by a 2x2 choice of subject and object position. Natural languages allow more complex argument structure profiles, including canonical orders in which the verb and object are separated (VSO and OSV), or variability from argument-drop or scrambling. How these properties are learned is an important question that we leave for future work.

	English	French
Corpus	Brown: Eve	Lyon
# Children	1	5
Ages	1;6-2;31	1;0-3;0
# Words	81,687	885,334
# Utterances	14,232	182,511

Table 1: Corpora of child-directed English and French

English	French
0.36 np v	0.48 np v
0.20 v	0.21 np v np
0.20 np v np	0.13 v
0.17 v np	0.05 np np v
0.04 np v np np	0.03 np v np np
0.03 v np np	0.03 v np

Table 2: Proportions of most frequent string types

is a Dirichlet distribution with parameters $\vec{\alpha}^{A_G}$. We begin with the assumption that all components $\alpha_i^{A_G}$ are equal to 1, resulting in a uniform prior distribution, i.e. the model considers all possible expansions for A with equal probability.

4.2 Inference

From the observed strings, the model infers the posterior distribution over all grammars in its hypothesis space, $P(G \mid \vec{w})$. Calculating this posterior analytically would require marginalizing over both $\vec{\theta^G}$ and $\vec{t}$— i.e., integrating over the rule weights and summing over all possible trees for a string, for all strings in the data. This calculation is intractable. So, instead of marginalizing over all of the information in $\vec{t}$, we marginalize over only some of it, and sample the remaining partial analyses. We call these partial analyses "coarse structures" ($\vec{s}$), described below. We begin by randomly initializing a set of possible coarse structures for the observed strings. Then, we use Gibbs sampling to jointly infer the posterior $P(G, \vec{s} \mid \vec{w})$, alternating between sampling a new grammar according to $P(G \mid \vec{s}, \vec{w})$, and sampling new coarse structures according to $P(\vec{s} \mid G, \vec{w})$. This process will converge to the joint posterior distribution over G and $\vec{s}$.

The coarse structures $\vec{s}$ take the same shape as the trees generated by the learner's grammars, but abstract away from the distinction between core and noise rewrites in those trees. This corresponds to abstracting away from the distinction between solid and dashed lines in Fig. 2. Unlike a full tree, which commits to particular core vs. noise distinctions and therefore is compatible with only some grammars, any coarse structure is consistent with all of the grammars in the learner's hypothesis space: it might be generated by core rules in certain grammars, or by some combination of noise and core rules, or by *only* noise rules, which are shared across all grammars. Therefore, for every grammar G, $P(G \mid \vec{s}, \vec{w})$ is always non-zero, allowing us to draw samples from this posterior in a feasible way. We sample $\vec{s}$ from the posterior $P(\vec{s} \mid G, \vec{w})$ with a Hastings proposal, using a variant of an algorithm introduced by Johnson et al. (2007) and marginalizing over $\vec{\theta^G}$. See Appendix B for details.

5 Simulations

We tested our model on English and French. These languages are both canonically SVO, but differ in how strictly they adhere to this canonical pattern: English has rigid word order, whereas French allows a greater degree of argument dislocation. We show that our model sucessfully identifies SVO as the target grammar for its noisy data, and does so even in an expanded hypothesis space that allows a choice among more flexible discrete hypotheses. Moreover, our model out-performs a learner whose grammar allows all word-order rules with some probability (Fig. 3B), with a numerical bias to prefer rule weights that are close to 0 or 1. This shows that for this case study, our model fares better than the more common type of explicit regularization bias in prior literature.

5.1 Data

We used the Eve and Lyon CHILDES corpora (Brown, 1973; Demuth and Tremblay, 2008), which contain speech directed to English- and French-learning 1- and 2-year-olds (see Table 1). We searched these corpora for strings of one v and any number of satellite np's. We used a noisy heuristic to approximate the knowledge of infants at 15 months and younger, who can use functional cues— determiners, pronouns, and auxiliaries— to differentiate nouns and verbs (Babineau et al., 2020; Shi and Melançon, 2010; Hicks et al., 2007). We categorized any full pronoun as an np; any word following a determiner as the head of an np; and any word following an auxiliary as a v. *Wh*-words and object clitics were not categorized as np's, because they may not be recognized as such by infants learning basic word order (Perkins and Lidz, 2021; Brusini et al., 2017). Object clitics that are homophonous with determiners were treated erroneously as determiners, to simulate the uncertainty that infants might have about their category.

To create the datasets for our learner, we sampled 50 strings in their relevant proportions in each language (see Table 2). Over 30% of the strings in each language are incompatible with the core rules of the target SVO grammar. As a whole, these data cannot be generated by the core rules of any single grammar in the learner's hypothesis space, without considering the option of noise.

5.2 Results: Our model

Fig. 4 displays our model's inferred posterior probability distribution over the four Mixture PCFGs in its hypothesis space, averaged over 10 runs of the model in each language. In both English and French, the SVO grammar was assigned a higher posterior probability than any other grammar in the learner's hypothesis space (all $ps < 0.001$, Binomial tests). This shows that the learner's filtering mechanism allowed it to overcome the large amount of noise in its data. The learner successfully discovered that the best explanation for its data involved identifying some portions that were signal for core SVO word order, and some portions that came from noise processes.

5.3 Comparison: Fully-flexible model

In order to assess how much our model's success depended on a choice of discrete canonical word-order grammars, we constructed a comparison learner whose hypothesis space collapses the distinction between canonical and non-canonical structures. This "fully-flexible" hypothesis space consists of a single standard PCFG comprising all of the word-order rules across our learner's four grammars (Fig. 3B). For this model, learning canonical word order would mean identifying that some of its rules have probabilities near zero.

We tested two variants of this model. The first assumes that all rules in its hypothesis space are equally probable *a priori*, as in our original model. The second is numerically biased to regularize its rule weights, following the regularization approach in prior literature (Reali and Griffiths, 2009; Culbertson et al., 2013; Ferdinand et al., 2019). This regularization bias takes the form of a skewed prior over the rule weights $\vec{\theta}$ in the learner's grammar. For each nonterminal A, we set all component parameters α_i^A of the model's Dirichlet prior to a small value, 0.001. This biases the learner to put probability mass on only one expansion of a given nonterminal, and push the probabilities of other expansions towards zero.

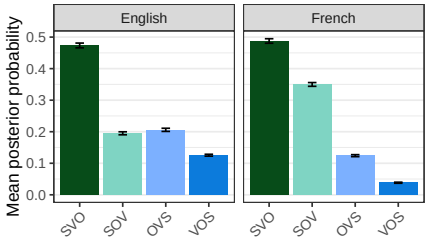


Figure 4: Posterior distribution over grammars

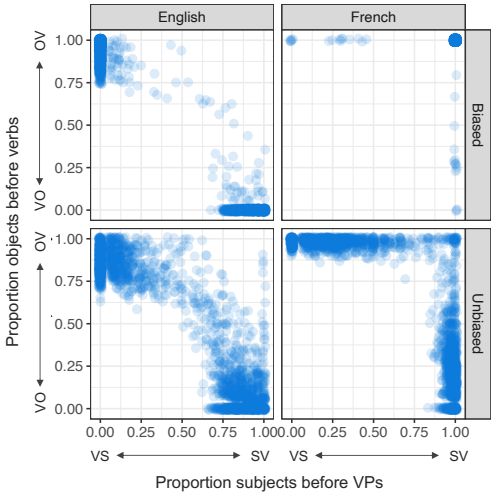


Figure 5: Posterior distribution over subject and object position in sampled treesets ($\vec{t}$), fully-flexible learner

The learner's inference process consists of one of the steps in our original Gibbs sampler. We sample trees for the learner's data from the posterior given its sole grammar, $P(\vec{t} \mid G, \vec{w})$, just as we sampled $P(\vec{s} \mid G, \vec{w})$ in our original model.

We assessed whether the fully-flexible learner had identified a canonical word order by calculating the proportion of the learner's sampled trees that contained subject NPs before verb phrases and object NPs before verbs. These proportions are plotted in Fig. 5, where each point corresponds to a sampled set of trees, aggregated across ten runs of the model in each language. These plotted distributions provide an estimate of the learner's inferred posterior probabilities of subject-initial and object-initial structures. The four possibilities for canonical word order correspond approximately to the four corners in each panel: clockwise from top left, these are OVS, SOV, SVO, and VOS.

If the learner had successfully identified that English and French are canonically SVO, the majority of tree samples would lie close to the lower right corners of these graphs. Instead, the unbi-

ased learner (bottom) inferred a distribution over tree structures that mirrored its noisy data. These ranged from the OVS to the SVO regions in English, and across the OVS, SOV, and SVO regions in French. The biased learner (top) inferred distributions closer to the corners corresponding to canonical word orders. However, the English learner gave equal posterior probability to both OVS and SVO structures; its mean proportions of subject-initial and object-final trees were not significantly different from 0.5 (mean subject-initial: 0.51, mean object-final: 0.54, $ps > 0.67$). The French learner converged to SOV structures instead of SVO (mean subject-initial: 0.99, mean subject-final: 0.01, $ps < 0.001$). The learner's regularization bias helped it identify one or two canonical word orders for its noisy data. But unlike our model, it did not correctly converge on SVO as the most probable word order in either language.

Why would our approach fare better than the more common approach to regularization in past work? Our model's success comes in large part from its expectation that canonical clauses require subjects; subject-drop can occur only in non-canonical clause types. This allows our learner to use the large number of np v strings as evidence for a subject-initial grammar. Given the choice between using its restricted core rules to analyze the sole np as a canonical subject, versus using its noise rules to analyze the np in a different position, a preference emerges for the canonical-subject analysis— just as we prefer to analyze a sequence of heads as coming from a two-headed rather than a head-tail coin. The fully-flexible learner does not distinguish between canonical structures in which subjects are required, and non-canonical structures in which they are not, so no preference emerges to to analyze a sole np in a specific clausal position.

For this learning problem, it appears helpful to have a hypothesis space with a distinction between core rules that provide deterministic options for canonical word order, and noise rules that produce non-canonical structures. This mixture of deterministic and non-deterministic options is what allows the target basic clause structure to emerge as the best explanation of the learner's noisy data.

5.4 Comparison: A data-coverage heuristic

Our learner successfully shows a preference for some hypotheses over others in a scenario where none are compatible with all of the data. But

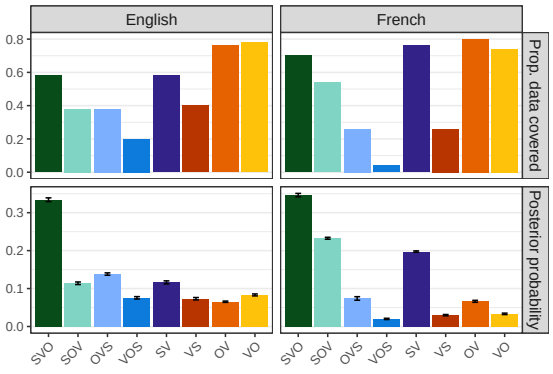


Figure 6: Eight-way hypothesis space: proportion data coverage vs. model's posterior distribution

one might ask whether the same result could be achieved via a much simpler approach: the core rules of the SVO grammar can generate 56% of the English data, which is a greater proportion than can be generated by the core rules of any of the alternatives (each less than 40%), and so this alone might lead a learner to identify SVO as the preferred option. Our model's inference mechanism does more than simply recapitulate this "data coverage" heuristic. To see this, it is useful to consider a scenario where the learner has a wider range of discrete hypotheses to choose among, including some that are more restrictive than others.

We constructed a comparison learner that considers an eight-way choice among Mixture PCFGs. These include all four deterministic options from our original model: grammars whose core rules fix subject and object position. In addition, the hypothesis space includes four more flexible grammars whose core rules fix only one of those argument positions, and allow the other to vary. For instance, the grammar we call "SV" fixes the subject preverbally, but allows the object to appear either before or after the verb: its rules are the union of the rules in the SVO and SOV grammars. Similarly, the "VS" grammar fixes the subject post-verbally but allows object position to vary, and the "OV" and "VO" grammars fix object position, but allow subject position to vary. All eight grammars share the same set of noise rules as in our original learner.

Given a choice among these eight grammars, the data-coverage heuristic will always favor one of the four more flexible ones, since they generate unions of the stringsets generated by the original four. In each of the top panels in Fig. 6, where a comparison only among the leftmost four grammars would have SVO as the winner (roughly mirroring Fig. 4),

we see that the more flexible grammars in general fare better by the data-coverage metric. But our learner, on both languages, assigned SVO higher posterior probability than any other grammar in the hypothesis space (Fig. 6, bottom; all $ps < 0.001$, averaged across 10 runs of the learner).

Why does our learner still succeed at identifying that English and French are SVO, even when there are other hypotheses that cover more of the data? Intuitively, our learner considers a tradeoff between fit to the data and restrictiveness of its hypotheses. Given the choice between the restrictive SVO hypothesis that provides a decent fit to the data, and the more flexible hypotheses that provide slightly better fits, a preference emerges for the more restrictive option— again paralleling our intuitive preference to attribute as many coin flips as possible to a two-headed rather than a head-tail coin. In our original model, this preference for restrictive hypotheses applied *within* each grammar, governing the learner's choice of attributing data to the restrictive core rules vs. the flexible noise rules. Here, we show that this same mechanism informs the learner's choice *across* grammars.

These findings demonstrate the flexibility and robustness of this learning mechanism. Our learner identifies strict SVO word order as its preferred hypothesis not only in comparison with other equally-strict alternatives, but also when other less restrictive options are available; the fact that it settled on deterministic SVO order in Fig. 4 was not simply a by-product of the fact that we provided only deterministic options. An implicit tradeoff between a grammar's restrictiveness and its fit to the data, and the expectation that this fit will be noisy, together enable the learner to identify the target deterministic word order among more flexible hypotheses.

6 Discussion

We introduce a general mechanism for noise-tolerant learning of deterministic grammars. Our learner assumes that its data are generated by a complex system: the particular grammatical processes that the learner is currently trying to acquire, and other unknown processes that conspire to introduce variability into the data. We model the inference process as a special case of probabilistic grammar learning, in which the learner evaluates a choice among different *Mixture PCFGs*: composite grammars in which each node might be introduced either by a restricted set of "core" rules, or by a less restricted set of "noise" rules.

We apply this approach to the problem of acquiring basic word order from immature sentence representations. Using distributions of imperfectly-identified noun phrases and verbs, our model successfully infers that English and French are SVO, without further cues to underlying sentence structure. It does so by separating signal for canonical word order from noise due to non-canonical structures, thereby implementing a proposal that young learners "filter" non-canonical clauses from their data (Pinker, 1984; Perkins et al., 2022). Because the learner's grammatical hypotheses allow only certain restricted core rules, a preference emerges to use these core rules to explain the skews in its data when possible, rather than analyzing most of the data as noise. This provides the impetus for successful filtering, even though our learner does not know ahead of time the rate or properties of non-canonical clauses in the language.

While we focus here on Mixture PCFGs, this same approach can be applied to "mixture" versions of other sorts of grammars that generate complex structures as a function of local choices about smaller subparts. This approach may therefore generalize to many other problems in grammar learning: e.g., learning phonological constraints that can be expressed in mixture finite-state systems, or learning syntactic dependencies that can be expressed in mixture multiple context-free grammars.

More broadly, this approach provides a novel mechanism for regularization in grammar learning. Here, a learner's tendency to regularize variable data is not driven by an explicit bias to prefer extreme points in a fully-gradient space, but instead emerges from the learner's expectation that its data are a noisy realization of a restrictive underlying system. This invites the possibility that other observed cases of regularization may be accounted for without adopting a fully-flexible hypothesis space. Instead, successful learning in certain domains may be underwritten by deterministic options in the learner's hypothesis space, combined with a general mechanism for filtering signal from noise.

Acknowledgments

We thank Xinyue Cui, Naomi Feldman, Jeff Lidz, Shalinee Maitra, the audiences at BUCLD 2022 and the UCLA Psycholinguistics/Computational Linguistics Seminar, and two anonymous reviewers for helpful feedback and assistance.

References

Mireille Babineau, Rushen Shi, and Anne Christophe. 2020. 14-month-olds exploit verbs' syntactic contexts to build expectations about novel words. *Infancy*, 25(5):719–733. Publisher: Wiley Online Library.

Roger Brown. 1973. *A First Language: The Early Stages*. Harvard University Press, Cambridge, MA.

Perrine Brusini, Ghislaine Dehaene-Lambertz, Marieke Van Heugten, Alex De Carvalho, François Goffinet, Anne-Caroline Fiévet, and Anne Christophe. 2017. Ambiguous function words do not prevent 18-month-olds from building accurate syntactic category expectations: An ERP study. *Neuropsychologia*, 98:4–12. Publisher: Elsevier.

Anne Christophe, Séverine Millotte, Savita Bernal, and Jeffrey Lidz. 2008. Bootstrapping Lexical and Syntactic Acquisition. *Language and Speech*, 51(1-2):61–75.

Jennifer Culbertson, Paul Smolensky, and Colin Wilson. 2013. Cognitive biases, linguistic universals, and constraint-based grammar learning. *Topics in Cognitive Science*, 5(3):392–424.

Katherine Demuth and Annie Tremblay. 2008. Prosodically-conditioned variability in children's production of French determiners. *Journal of child language*, 35(1):99–127. Publisher: Cambridge University Press.

Vanessa Ferdinand, Simon Kirby, and Kenny Smith. 2019. The cognitive roots of regularization in language. *Cognition*, 184:53–68.

Lila Gleitman. 1990. The structural sources of verb meanings. *Language Acquisition*, 1:3–55.

Jessica Hicks, Jessica Maye, and Jeffrey Lidz. 2007. The role of function words in infants' syntactic categorization of novel words. In *Proceedings of the Linguistic Society of America Annual Meeting*, Anaheim, CA.

Kathy Hirsh-Pasek and Roberta Michnick Golinkoff. 1996. The intermodal preferential looking paradigm: A window onto emerging language comprehension. In Dana McDaniel, Cecile McKee, and Helen S. Cairns, editors, *Methods for assessing children's syntax*, pages 105–124. The MIT Press, Cambridge, MA.

Carla L. Hudson Kam and Elissa L. Newport. 2005. Regularizing Unpredictable Variation: The Roles of Adult and Child Learners in Language Formation and Change. *Language Learning and Development*, 1(2):151–195.

Carla L. Hudson Kam and Elissa L. Newport. 2009. Getting it right by getting it wrong: When learners change languages. *Cognitive Psychology*, 59(1):30–66.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146, Rochester, New York. Association for Computational Linguistics.

Jeffrey Lidz, Aaron Steven White, and Rebecca Baier. 2017. The role of incremental parsing in syntactically conditioned word learning. *Cognitive Psychology*, 97:62–78.

Laurel Perkins, Naomi H. Feldman, and Jeffrey Lidz. 2022. The Power of Ignoring: Filtering Input for Argument Structure Acquisition. *Cognitive Science*, 46:e13080.

Laurel Perkins and Jeffrey Lidz. 2020. Filler-gap dependency comprehension at 15 months: The role of vocabulary. *Language Acquisition*, 27(1):98–115.

Laurel Perkins and Jeffrey Lidz. 2021. 18-month-old infants represent non-local syntactic dependencies. *Proceedings of the National Academy of Sciences*, 118(41):e2026469118.

Steven Pinker. 1984. *Language learnability and language development*. Harvard University Press, Cambridge, MA.

Florencia Reali and Thomas L. Griffiths. 2009. The evolution of frequency distributions: Relating regularization to inductive biases through iterated learning. *Cognition*, 111:317–328.

Jordan Schneider, Laurel Perkins, and Naomi H. Feldman. 2020. A noisy channel model for systematizing unpredictable input variation. In *Proceedings of the 44th Annual Boston University Conference on Language Development*, pages 533–547.

Rushen Shi and Andréane Melançon. 2010. Syntactic Categorization in French-Learning Infants. *Infancy*, 15(5):517–533.

A Details of the coins example from Section 2

Recall the scenario with just Bag H: this bag contains an unknown number of Type A coins, which always come up heads, and an unknown number of Type B coins, which all have some single unknown probability θ of coming up heads. Ten times, a coin is chosen from the bag and flipped; this produces eight heads and two tails. How many of these ten flips might we guess came from Type A coins, and how many from Type B coins?

We consider three hypotheses:

- H1: 0 Type A flips, 10 Type B flips

- H2: 6 Type A flips, 4 Type B flips
- H3: 8 Type A flips, 2 Type B flips

The three hypotheses' likelihoods, conditioned upon the unknown probability θ, are as follows.

$$(4)\quad \Pr(\text{data} \mid \text{H1}, \theta) = \binom{10}{8}\, \theta^8\, (1-\theta)^2$$

$$(5)\quad \begin{aligned}\Pr(\text{data} \mid \text{H2}, \theta) &= \binom{6}{6}\, 1^6 \cdot \binom{4}{2}\, \theta^2\, (1-\theta)^2 \\ &= \binom{4}{2}\, \theta^2\, (1-\theta)^2\end{aligned}$$

$$(6)\quad \begin{aligned}\Pr(\text{data} \mid \text{H3}, \theta) &= \binom{8}{8}\, 1^8 \cdot \binom{2}{0}\, \theta^0\, (1-\theta)^2 \\ &= (1-\theta)^2\end{aligned}$$

As noted in the main text, these expressions highlight the fact that H1 is the most costly hypothesis, since it relies most heavily on the contingent outcomes from Type B coins, and H3 is the least costly.

We can make this more precise by marginalizing over the unknown value of θ in (4) and (5). The useful general result here is that

$$(7)\quad \int_0^1 \binom{n}{k} \theta^k (1-\theta)^{n-k} d\theta = \frac{1}{n+1}$$

for any n and k; notice that the right-hand side only depends on n. Marginalizing over θ in (4), (5) and (6), under the assumption of a uniform prior on θ, yields integrals of this form.[5] For H1, $n = 10$ so $\Pr(\text{data} \mid \text{H1}) = \frac{1}{11}$. What this highlights is that the likelihood under such a hypothesis depends *only* on the number of times that hypothesis needs to invoke the uncertain Type B coin flip: *any* outcome of the ten-flip experiment invoked by H1 has probability $\frac{1}{11}$, and *any* outcome of the two-flip experiment invoked by H3 has probability $\frac{1}{3}$.

$$\begin{aligned}\Pr(\text{data} \mid \text{H1}) &= \tfrac{1}{11} \\ \Pr(\text{data} \mid \text{H2}) &= \tfrac{1}{5} \\ \Pr(\text{data} \mid \text{H3}) &= \tfrac{1}{3}\end{aligned}$$

Now consider the choice between Bag H and Bag T, as candidate explanations for a sequence of ten flips that yielded eight heads and two tails. We have seen that, using Bag H, the possible hypotheses range from those that provide "good" explanations of the data (such as H3 at one extreme) by exploiting the presence of the two-headed coins, to those that constitute "costly" explanations (such as H1 at the other extreme) because they rely heavily on flips of the head-tail coins; see Fig. 7. With Bag T, the explanations at the costly extreme are still available (e.g. the hypothesis that all ten flips came from head-tail coins; $n = 10$), but there is no way for the two-tailed coins to contribute to particularly good explanations of the observed high proportion of heads. To minimize the reliance on the contingent outcomes of head-tail coins, the best one can do is to suppose (call this H4) that the two observed tails both came from two-tailed coins, which still leaves eight uncertain flips. The likelihood under this hypothesis (compare with (5) for H2) is

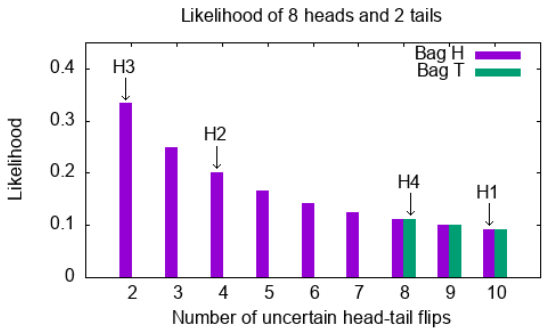


Figure 7

$$(8)\quad \begin{aligned}\Pr(\text{data} \mid \text{H4}, \theta) &= \binom{2}{2}\, 1^2 \cdot \binom{8}{0}\, \theta^8\, (1-\theta)^0 \\ &= \theta^8\end{aligned}$$

and $\Pr(\text{data} \mid \text{H4}) = \frac{1}{9}$.

Returning now to the overarching choice between the two bags: the likelihood assigned to the data by a particular bag is the sum of the heights of the associated bars in Fig. 7. This is clearly larger for Bag H, and so assuming a flat prior over the two bags, the posterior probability of Bag H will be higher than that of Bag T.

B Details of Gibbs sampling

In the first step of sampling, we use Bayes' Rule to calculate the posterior probability of each grammar given the observed strings $\vec{w}$ and a collection of hypothesized coarse structures $\vec{s}$ for those strings:

$$(9)\quad P(G|\vec{s},\vec{w}) = \frac{P(\vec{s},\vec{w}|G)P(G)}{\sum_{G'} P(\vec{s},\vec{w}|G')P(G')}$$

Bayes' Rule tells us that the posterior probability of any grammar is proportional to the product of

[5] Specifically, the uniform prior can be represented as a Beta(1,1) distribution over θ, so $\Pr(\text{data} \mid \text{H1}) = \int_0^1 \Pr(\text{data} \mid \theta, \text{H1})\text{Beta}(\theta \mid 1,1)d\theta = \int_0^1 \Pr(\text{data} \mid \theta, \text{H1})d\theta$, since $\text{Beta}(\theta \mid 1,1) = 1$ for all θ.

the likelihood (the probability of $\vec{s}$ and $\vec{w}$ under that grammar) and the prior probability of that grammar. We assume that all four grammars have equal prior probability.

Because we are only considering coarse structures that could have yielded the strings in the data, the joint likelihood of the coarse structures and strings, $P(\vec{s}, \vec{w}|G)$, is equivalent to the likelihood of the coarse structures alone, $P(\vec{s}|G)$. Calculating this likelihood requires summing over the unknown ways that each portion of these coarse structures might be analyzed as either a core (ϕ, solid line) or noise (ψ, dashed line) rewrite. The specific core vs. noise choices are interchangeable for each particular nonterminal given a grammar, so we make this calculation tractable by considering how *many* core vs. noise rewrites might have occurred for each nonterminal.

We divide the n^A total observations of a particular nonterminal A into $n_1^A \ldots n_m^A$ observations of the 1^{st} through the m^{th} possible rewrites (collapsing across ϕ-rewrites and ψ-rewrites of A). The full likelihood of the set of coarse structures, $P(\vec{s}|G)$, is the product over all nonterminals A of $P(n_1^A \ldots n_m^A \mid G)$. We divide each of the observed rewrites of a nonterminal into some number of core (solid line) rewrites (ϕ) and some number of noisy (dashed line) rewrites (ψ). The n_1^A occurrences of the first type of rewrite for A are divided into $n_1^{A^\phi}$ core occurrences and $n_1^{A^\psi}$ noisy occurrences. More generally, the n_m^A occurrences of the m^{th} rewrite type are divided into $n_m^{A^\phi}$ core occurrences and $n_m^{A^\psi}$ noisy occurrences. We can calculate the likelihood by marginalizing over $n_1^{A^\phi} \ldots n_m^{A^\psi}$:

$$P(\vec{s}|G) = \prod_A P(n_1^A \ldots n_m^A \mid G) = \tag{10}$$

$$\prod_A \Bigg[\sum_{n_1^{A^\phi}=0}^{n_1^A} \cdots \sum_{n_m^{A^\phi}=0}^{n_m^A} \Big[P(n_1^{A^\phi} \ldots n_m^{A^\phi} | n^{A^\phi}, G) \times P(n_1^{A^\psi} \ldots n_m^{A^\psi} | n^{A^\psi}, G) \times P(n^{A^\phi} | n^A, G) \Big] \Bigg]$$

The first term in the sum is the probability of observing $n_1^{A^\phi} \ldots n_m^{A^\phi}$ core occurrences of each rewrite type, out of n^{A^ϕ} total core occurrences of A. This follows a multinomial distribution with parameter $\vec{\phi}^{A_G}$. Because $\vec{\phi}^{A_G}$ is unknown, we integrate over all possible values of $\vec{\phi}^{A_G}$ to obtain

$$\frac{B(\vec{\alpha_\phi}^{A_G} + (n_1^{A^\phi} \ldots n_m^{A^\phi}))}{B(\vec{\alpha_\phi}^{A_G})} \tag{11}$$

for this first term, where $\vec{\alpha_\phi}^{A_G}$ represents the parameters of the Dirichlet prior over $\vec{\phi}^{A_G}$, and $B(\cdot)$ is the multivariate Beta function.

The second term in the sum in (10) is analogous: this is the probability, given n^{A^ψ} total noisy occurrences of A, of observing $n_1^{A^\psi} \ldots n_m^{A^\psi}$ noisy occurrences of each rewrite type, which follows a multinomial distribution with parameter $\vec{\psi}^{A_G}$. The third term is the probability of observing n^{A^ϕ} total core occurrences out of n^A overall occurrences of A. This follows a binomial distribution with parameter $(1 - \epsilon^{A_G})$. We again integrate over all possible values of $\vec{\psi}^{A_G}$ and ϵ^{A_G}, obtaining results analogous to Eq. (11).

This allows us to calculate the likelihood $P(\vec{s} \mid G)$ for each G in our hypothesis space, and (since we assume a flat prior of grammars) sample a new G with probability proportional to this likelihood.

After re-sampling a new grammar G, we then use a component-wise Hastings proposal to sample a new set of coarse structures $\vec{s}$ for the observed strings, given G. Following Johnson et al. (2007), we consider the probability of a particular coarse structure s_i for corresponding string w_i, given G and the current hypotheses about coarse structures $\vec{s}_{-i}$ for all the other strings. We can define a function f that is proportional to the posterior distribution over s_i, $f(s_i) \propto P(s_i|w_i, \vec{s}_{-i}, G)$, as

$$f(s_i) = P(w_i|s_i)P(s_i|\vec{s}_{-i}, G) \tag{12}$$

The probability of a string being the yield of a given coarse structure, $P(w_i|s_i)$, is always 1 or 0. The probability of a coarse structure given all other coarse structures and G, $P(s_i|\vec{s}_{-i}, G)$, is

$$P(s_i|\vec{s}_{-i}, G) = \frac{P(\vec{s}|G)}{P(\vec{s}_{-i}|G)} \tag{13}$$

Both $P(\vec{s}|G)$ and $P(\vec{s}_{-i}|G)$ are calculated according to to Eq. (10).

We can use this function f to sample $\vec{s}$ given G and $\vec{w}$ as follows. Within each iteration of the Gibbs sampler, we re-sample $\vec{s}$ using a procedure modified from Johnson et al. (2007). First, we choose a string w_i and its current corresponding s_i at random. Second, we take the other coarse structures $\vec{s}_{-i}$, to be the output of a simple PCFG

which generates coarse structures directly, rather than the full trees generated by a Mixture PCFG. We estimate of the weights of this PCFG, $\vec{\theta^s}$, from the relative frequencies of each observed rewrite, using add-one smoothing to account for accidental gaps. Third, we generate a new proposed coarse structure $s_i{}'$ for w_i by sampling from this grammar's distribution using $\vec{\theta^s}$. Finally, we decide to accept this proposal with probability

$$A(s_i') = min\left(1, \frac{f(s_i{}')P(s_i|w_i, \vec{\theta^s})}{f(s_i)P(s_i{}'|w_i, \vec{\theta^s})}\right) \quad (14)$$

We ran multiple chains from different starting places to test convergence. For the simulations reported in Sec. 5.2, we ran chains of 20,000 iterations of Gibbs sampling each, and analyzed every 10th iteration from the last half of each chain. We report averages across 10 chains as estimates of the posterior over G and $\vec{s}$. To simulate the "fully-flexible" learner described in Sec. 5.3, we estimate the posterior distribution over $\vec{t}$ by using a component-wise Hastings sampler analogous to that for estimating $P(\vec{s}|G, \vec{w})$ in our original model. We ran 10 chains of 20,000 Hastings iterations each, and analyzed every 10th iteration from the last half of each chain.

On the Spectra of Syntactic Structures

Isabella Senturia
Yale University
isabella.senturia@yale.edu

Robert Frank
Yale University
robert.frank@yale.edu

Abstract

This paper explores the application of spectral graph theory to the problem of characterizing linguistically significant classes of tree structures. As a case study, we focus on three classes of trees, binary, X–bar, and asymmetric c–command extensional, and show that the spectral properties of different matrix representations of these classes of trees provide insight into the properties that characterize these classes. More generally, our goal is to provide another route to understanding the structure of natural language, one that does not come from extensive definitions and rules taken by extrapolating from the syntactic structure, but instead is extracted directly from computation on the syntactically–defined graphical structures.

1 Introduction

In order to explore properties of natural and artificial language, the choice of representation is extremely important, as one is constrained to work within the tools existent for that representation. Motivated by immediate consituency theory, tree–structured graphical representations are the overwhelming favorite of syntacticians, capturing the multidimensionality inherent in the hierarchical structures of grammar. Modern graphical representations of syntax utilize binary trees: rooted tree graphs where each node branches into 0, 1 or 2 new nodes.

Syntacticians ask what constraints exist on tree structures by deriving properties of the acceptable structure and extrapolating from those potential rules and axioms governing natural language structures. All syntactic trees are rooted and downward branching. The most basic of restrictions syntacticians have imposed on a syntax tree is the branching factor of the nodes: it is widely assumed that syntactic trees are binary branching.

Another attempt by syntacticians to constrain permissible tree structures which accurately model natural language is *X–bar theory*: all phrases require the template of XP branching into specifier SpecXP and X′, and X′ branching into head X and complement CompX, as in Figure 1. SpecXP and CompX are optional—if they do not exist, neither do the edges connecting them to the structure (denoted by the dashed lines). If they do exist, they themselves have to follow the same structural guidelines of X–bar theory.

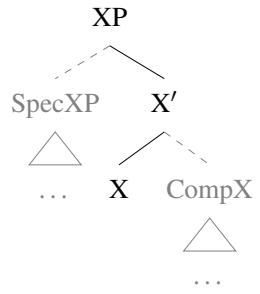


Figure 1: The requisite underlying structure of a phrase XP.

Kayne (1994) develops another restriction on possible tree structures by means of the Linear Correspondence Axiom (LCA), which states that the asymmetrical c–command relationship is a strict linear order (i.e. irreflexive, transitive, and asymmetric). Well–formed versus ill–formed trees can then be characterized as a result of the hierarchy (by way of the LCA and asymmetric c–command).

Frank and Vijay-Shanker (2001) suggest a partial order defined by a c–command relation as a primitive relation and that which should determine the hierarchy of syntactic tree structures (as opposed to dominance, by deriving dominance using the c–command relation). Frank and Kuminiak (2000) extended this idea to *asymmetric c–command*, suggesting that asymmetric c–command is a primitive relation, defining trees using this relation and arguing that this class is very similar to X–bar trees. Kuminiak (1999) considers classes of trees that are uniquely definable by some relation—more specifically, those that are uniquely defined by their asymmetric c–command relation, i.e. *asymmetric*

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 199-211.
Amherst, Massachusetts, June 15-17, 2023

c–command extensional (ACC).

Much of the work studying constraints on syntactic structures that accurately reflect properties of natural language has been done in a vein similar to the aformentioned work, by way of thinking about which structures are syntactically valid, and then attempting to generalize these properties. This paper provides an alternate route, one which directly studies syntactic classes from a mathematical perspective. While many properties are not derivable directly from the graphical structure, the aformentioned work demonstrates some which are. This paper explores the three previously–defined classes of trees—binary, X–bar, and ACC—from the vantage point of spectral graph theory.

Spectral graph theory (SGT) maps graphs to various matrix representations and analyzes spectral properties of these matrices. [1] Simple eigenvalue/eigenvector properties of a graph's matrix can be linked to properties of the graph that are often of high importance to the mathematician/computer scientist, such as graph–coloring and graph isomorphism (Wilf, 1967; Hoffman, 1970; Spielman, 2019; Chung, 1997; Godsil and Royle, 2001).

Researchers explore the distribution of eigenvalues of various graphs across the real numbers and concrete bounds on these distributions. A host of work explores whether graphs can be determined or distinguished by their spectra: cf. van Dam and Haemers (2003), Haemers and Spence (2004).

The notion of a tree has long existed within the mathematical subfield of graph theory, and trees have been extensively studied within both graph theory and spectral graph theory. Jacobs et al. (2021) study the distribution of eigenvalues of tree graphs. Dadedzi (2018) analyzes the spectra of various classes of trees, developing bounds on multiplicities of eigenvalues. Work has been done studying the spectrum of $k-$ary trees, trees where every non-leaf node has *branching factor*, i.e. degree, of $k \in \mathbb{N}$, and each leaf has degree 1 (He et al., 2000; Wang and Xu, 2006).

With respect to linguistic questions, Chowdhury et al. (2021) demonstrates an application of SGT to phylogenetic trees involving different graph isomorphism techniques. Ortegaray et al. (2021) use eigenvectors of the Laplacian matrix to detect relations between various vectors of syntactic parameter values.

SGT has not, however, been used to explore graphical properties of linguistic classes of tree structures. This paper demonstrates the utility in doing just that. It presents natural spectral properties of these trees that distinguish desirable classes of syntactic structures, exploring the extent to which these classes can be characterized by properties of their spectra.

The paper is structured as follows. Section 2 introduces the formal mathematical tools necessary: graph theory, matrix theory, and spectral graph theory. Section 3 explores spectral properties of the undirected graphs, before pivoting to those properties of directed graphs in section 4. Section 5 concludes.

2 Mathematical preliminaries

We present the mathematical notations and concepts of the paper, beginning with graph theory.

2.1 Graph Theory

Formally, we define a graph $G = (V(G), E(G))$, where $V(G) = \{v_1, v_2, \ldots, v_n\}$ is a set of n *vertices*, $E(G) = \{\{v_a, v_b\}, \ldots, \{v_p, v_q\}\}$ is a set of m *edges*.[2] We often abbreviate this notation to $G = (V, E)$, and label a set of k nodes with integers 1 through k. If the edges are undirected, the edge pair $\{v_i, v_j\}$ is unordered, whereas if the edge is directed, the edge pair is ordered $\{start, end\}$.

The *degree* d_v of a vertex v is the number of edges connected to that node. For directed graphs, we use *outdegree*, the number of edges leaving that node. We denote the set of (out)degrees of a graph G as $\mathcal{D}(G)$. A *leaf* is a node of degree 1 (or, in the case of a *directed graph*, i.e. digraph, outdegree 0). Two *adjacent* vertices are connected by a single edge. A *quasipendant* vertex is a vertex adjacent to a leaf. A *path* from some vertex v_i to another v_j is the sequence of edges connecting adjacent nodes between v_i and v_j. A graph is *connected* if there is a path from every node to every other node.

Graphs are often divided into *classes*. Graphs in a given class have one or more (often structural) unifying characteristics. The class of *trees* is the class of connected *acyclic graphs* $T = (V, E)$ defined by the existence of exactly one path connecting any two given vertices $v_1, v_2 \in V$—that is, they have no loops. A *directed tree* is a tree with directed edges. A *rooted* tree is a tree for which a

[1] We thus interchangeably refer to the spectra of a matrix representing a graph as the spectra of the graph.

[2] We follow the presentation of graph theory of Bondy and Murty (1976).

specific node has been designated as the root, and is graphed with this root at the top or bottom. Any directed tree, i.e. directed acyclic connected graph, will have a root: the node that has no edges entering it.

2.2 Spectral Graph Theory

Mathematicians have explored different ways to represent graphs, outside of the canonical picture of nodes and edges. Spectral graph theory, exploring algebraic representations of graphs by mapping graphs to various matrix representations, provides an approach to both explore what sorts of graphical properties (already observable through the graph–theoretic depiction) can be captured algebraically, and what new otherwise–unperceived properties emerge by virtue of the algebraic representation.

Spectral graph theory explores the link between algebra and graph theory by examining algebraic properties of matrix representations of graphs and how they reflect or represent combinatorial properties of these graphs.[3] We construct a mapping from a graph $G = (V, E)$ to a matrix $M \in \mathbb{F}^n \times \mathbb{F}^n$, where $\mathbb{F}$ is the field over which the entries of M are defined[4] and m_{ij} contains information about v_i, v_j, or the edge connecting them. Shifting between two different mathematical representations, a graph and a matrix, of the same mathematical object, allows both graphical/combinatorial and algebraic exploration of this object, permitting discovery of connections across these subfields that can be used to capture otherwise unascertainable properties of the graph.

A number of possible matrix representations are available for graphs, including the adjacency matrix A_G and diagonal matrix D_G (McKay, 1977).

Definition 2.1. *Given a graph $G = (V, E)$, we define the entries of the* adjacency matrix $A_G \in \mathbb{N}^{|V|} \times \mathbb{N}^{|V|}$ *as follows:*

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

In the case of undirected graphs, the adjacency matrix will be symmetric (as $\{v_i, v_j\} \in E \iff \{v_j, v_i\} \in E$), whereas digraphs' adjacency matrices are not symmetric.

Definition 2.2. *Given $G = (V, E)$, let the* diagonal matrix $D_G \in \mathbb{N}^{|V|} \times \mathbb{N}^{|V|}$ *be defined as:*

$$d_{ii} = \sum_{j \in |V|} \mathbb{1}(\{v_i, v_j\}),$$

where the indicator function $\mathbb{1}(\{v_i, v_j\})$ is 1 when the edge $\{v_i, v_j\}$ exists, and 0 otherwise.

These d_{ii} values indicate the degree d_{vi} of each node v_i. So intuitively, D_G records the degree of each v_i in the i^{th} diagonal.

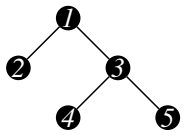
Given these two matrix representations of a graph, we can now define the Laplacian.

Definition 2.3. *Let $G = (V, E)$ be a graph with adjacency matrix A_G and diagonal matrix D_G. The* Laplacian *is defined as*

$$L_G = D_G - A_G$$

In the following example, we give an undirected binary tree with five nodes and construct its adjacency, diagonal and Laplacian matrix representations.

Example 2.4. *Consider the undirected rooted binary tree $G = (V, E)$ with $V = \{1, 2, 3, 4, 5\}$, $E = \{\{1, 2\}, \{1, 3\}, \{3, 4\}, \{3, 5\}\}$:*



$$A_G = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \; D_G = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$L_G = D_G - A_G = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

An (uncommon) variation on the Laplacian, the signless Laplacian, is also relevant to this paper.

Definition 2.5. *Let $G = (V, E)$ be a graph with adjacency matrix A_G and diagonal matrix D_G. The* signless Laplacian *is defined as*

$$\hat{L}_G = |L_G| = D_G + A_G$$

After mapping the graph to a matrix representation, such as the Laplacian, we have all the tools of linear algebra at our disposal.

[3] Spielman (2019), Chung (1997) and Godsil and Royle (2001) form the basis of the following discussion.

[4] In this paper, we deal with the field of real numbers $\mathbb{R}$.

2.3 Spectral Theory

Spectral graph theory is based in *eigentheory*, the theory of eigenvalues and eigenvectors of matrices.

Definition 2.6. *A vector $\psi \in \mathbb{R}^n$ is an* eigenvector of matrix $M \in \mathbb{R}^n \times \mathbb{R}^n$ with eigenvalue $\lambda \in \mathbb{R}$ *if it is nonzero and if*

$$M\psi = \lambda\psi$$

For any matrix M and vector v (of the proper dimensions), the product Mv indicates M acting as a linear transformation via scaling and rotation. However, for all eigenvalues λ (a scalar) of M and their corresponding eigenvectors ψ, the equation $M\psi = \lambda\psi$ signals the ψ are those vectors for which M does not rotate but only scales by a factor of λ.

A matrix of dimension n has n (not necessarily unique) eigenvalues. We follow the convention of denoting this set of eigenvalues of a graph G's matrix representation M_G, known as the *spectrum* of M_G, as $\Lambda(M_G) = \{\lambda_1, ..., \lambda_n\}$, where the eigenvalues $\lambda_1, ..., \lambda_n$ are ordered from smallest to largest (that is, $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$). The *multiplicity* of an eigenvalue λ in the spectrum of M, denoted $\mu_M(\lambda)$, is the number of times that λ occurs. Within $\Lambda(M)$, an eigenvalue λ with multiplicity k is represented as λ^k.

Obviously any matrix representation of a graph changes with node labeling, as the node labels determine the position of node information in the matrix. However, the spectrum is *invariant* under permutation of the rows and columns of the matrix, meaning any permutation of the rows and (the same) columns of M yielding M' has the property that $\Lambda(M) = \Lambda(M')$. Thus, the spectrum of a graph is a useful way to explore properties of a graph as *isomorphic* graphs (graphs which are identical with a relabeling of nodes) have the same spectrum.

Spectral graph theory explores properties of these eigenvalues which have been extracted from the matrix of a graph to uncover combinatorial properties of the graph.

3 Spectral properties of undirected syntactic structures

This paper concerns the spectral properties of three classes of potentially syntactically–relevant graphs: binary trees, X–bar trees, and ACC trees.

Because the mathematics of undirected trees has been more widely studied, we begin with studying syntactic structures as undirected graphs. This ignores a crucial aspect of the tree structure assumed in linguistics—namely, the presence of a root node, and the ordered relationship between pairs of nodes (i.e. dominance). We completely ignore the issue of precedence among nodes so that trees are encoded entirely on the basis of their hierarchical relationships.

3.1 Generating classes

First, we define the three classes of graphs representing the three syntactic classes of binary, X–bar and ACC trees. Let `bin_base` be the smallest non–empty binary tree with three nodes, i.e. the three–noded path graph where $d_v = 2$ for the root v. Let $(T_\alpha, T_\beta) \uparrow$ `bin_base` denote the simultaneous substitution of the trees T_α and T_β into the left and right leaves of `bin_base`, respectively. In what follows, we assume the trees to be unordered.

`Bin(n)` is the class of all binary (branching) trees with $n = 2k + 1$ nodes defined recursively as

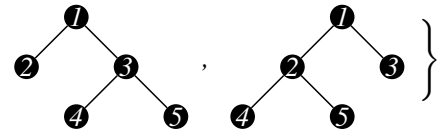
$$\texttt{Bin(2k + 1)} = \bigcup_{i=1}^{k-1} (T_\alpha, T_\beta) \uparrow \texttt{bin\_base}$$

over $T_\alpha \in \texttt{Bin(2i + 1)}, T_\beta \in \texttt{Bin(2k - 2i - 1)}$, where T_1 is `single_node`, the single–noded tree.

Example 3.1. *Let $T_\gamma =$* `single_node` *and $T_\delta =$* `bin_base`*. So* $\texttt{Bin(1)} = \{T_\gamma\}, \texttt{Bin(3)} = \{T_\delta\}$*, and*

$$\texttt{Bin(5)} = \{(T_\gamma, T_\delta) \uparrow T_\delta, (T_\delta, T_\gamma) \uparrow T_\delta\}$$

$$= \left\{ \quad , \quad \right\}$$



`Xbar(n)` is the class of all X–bar trees with $n = 3k$ nodes. Define the base `xbar` tree as the path graph with three nodes:[5]

$$\texttt{xbar} = (\{v_1, v_2, v_3\}, \{\{v_1, v_2\}, \{v_2, v_3\}\}).$$

Define two new substitution operations specific to this syntactic class, $T_\chi \uparrow^*_{spec}$ `xbar` and $T_\chi \uparrow^*_{comp}$ `xbar` as inserting T_χ into the specifier or complementizer of the base `xbar` tree by connecting the root of T_χ to the top/root (XP) node or middle (X′) node of `xbar`, respectively, with a new edge.

[5] This can be understood from Figure 1 as the path with nodes XP, X′, and X. As SpecXP and CompX are both empty, the edges denoted by dashed lines in 1 are also absent.

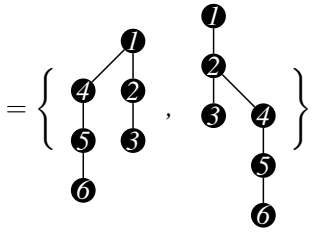
We denote by $(T_\chi, T_\rho) \uparrow^*$ xbar the simultaneous insertion of T_χ and T_ρ into the specifier and complementizer, respectively, of xbar.

$$\texttt{Xbar(3k)} = \bigcup_{i=1}^{k-1} (T_i, T_j) \uparrow^* \texttt{xbar}$$

for $T_i \in \texttt{Xbar(3i)}, T_j \in \texttt{Xbar(3(k - i))}$.

Example 3.2. $\texttt{Xbar(3)} = \{\texttt{xbar}\}$ *and*

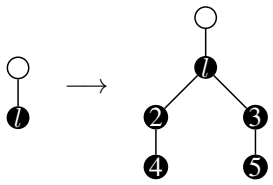
$\texttt{Xbar(6)} = \{\texttt{xbar} \uparrow^*_{spec} \texttt{xbar}, \texttt{xbar} \uparrow^*_{comp} \texttt{xbar}\}$



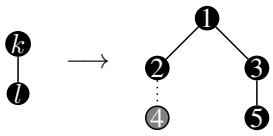
The natural interpretation of these are a single XP with a specifier of a single XP, and no complementizer, and a single XP containing a single complementizer of a single XP, and no specifier.

As presented by Kuminiak (1999), the asymmetric c–command extensional trees (i.e. those uniquely determined by their asymmetric c–command relation) can be generated by two types of insertion.

1. Add: Add two non–branching quasipendant vertices to any leaf.



2. Replace: For any nonbranching quasipendant node k, replace k with the five–noded structure below, with or without the left leaf node (4), i.e.



Then we can define the class ACC. Note we index families of trees from this class with number of *insertions*, as opposed to the number–of–node indexing we used previously, because each operation adds a variable number of nodes to the graph. We specify performing the Add (1) or Replace (2) operations at leaf node l (or in the case of the Replace operation (2), at l's quasipendant vertex, removing l altogether) as as $T_\alpha \uparrow^m_l T_\beta$, where T_0 is the empty tree, as

$$\texttt{ACC(k)} = \bigcup_{i=0}^{k} T_\alpha \uparrow^m_l T_\beta$$

for $T_\alpha \in \texttt{ACC(i)}, T_\beta \in \texttt{ACC(k - i)}, l \in L(T_j), m \in \{1, 2\}$.

Finally, we note a simple but important fact about the three defined classes.

Proposition 3.3. *For $n > 3$, the three classes are disjoint.*

When looking at the spectra of large trees from the three classes, this idea is useful in that it guarantees that the three tree sets are non–overlapping. So, it would be important for the spectra to reflect this fact.

3.2 Spectra of the three classes

It is known that the signless Laplacian spectrum and the Laplacian spectrum are identical for bipartite graphs (Abdian et al., 2018).[6] We additionally note that the magnitude of the eigenvectors of L_G and $\hat{L}_G$ are equal—the only difference stems from differences in sign in some of the entries of the vectors. Thus, we have the following proposition.

Proposition 3.4. *For any undirected rooted tree graph $T = (V, E)$ where $T \in$ BIN, XBAR, or ACC,*

$$\Lambda(L_G) = \Lambda(\hat{L}_G).$$

Further, the eigenvectors of L_G and $\hat{L}_G$ are identical modulo sign.

Now, we compare the spectra of the three classes of syntactic graphs by randomly generating three equal–sized sets (corresponding to the three syntactic classes) of high–dimensional[7] n–noded graphs, map them each to a matrix representation of dimension n, and graph their spectra in order of increasing value according to their percentile rank with the coordinates $(i \cdot \frac{100}{n}, \lambda_i)$.The trees are high–dimensional so the shape of the spectra is visible.

Each of the three graphs in Figure 2 demonstrate that each syntactic class has a unique spectrum distinct from the others: binary trees have the highest

[6] As such, the graph of $\hat{L}_G$ is omitted from this paper.

[7] We use "dimensional" to refer to the number of nodes in the graph, as the number of nodes in a graph corresponds to the number of dimensions of its matrix representations.

multiplicity of eigenvalues 0 and 1, followed by X–bar trees, while ACC trees are smoothest.

There are a couple of facts that help analyze the distribution of the spectrum. Let $l(T)$ be the number of leaves of a given tree, and $q(T)$ be the number of quasipendant vertices.

Corollary 3.5 (Nosal, 1970; Smith, 1970; Cvetkovic et al., 1980 p. 258.[8])**.** *The multiplicity of the eigenvalue* 0 *in the adjacency spectrum of a tree* T *is at least* $l(T) - q(T)$.

The same fact can be said of the eigenvalue 1 in the Laplacian spectrum:

Corollary 3.6 (Nosal, 1970; Smith, 1970; Cvetkovic et al., 1980 p. 258)**.** *The multiplicity of the eigenvalue* 1 *in the Laplacian spectrum of a tree* T *is at least* $l(T) - q(T)$.

It turns out that the multiplicity of eigenvalue 1 in the Laplacian spectrum, $\mu_L(1)$, is a tight lower bound for all three classes. For the binary trees, experimentation with randomly generated trees points to the number $l(T) - q(T)$ as either *exactly* $\mu_L(1)$, or 1 or 2 less than $\mu_L(1)$.[9] The few trees T generated experimentally whose multiplicity of eigenvalue 1 in the Laplacian is *not* equal to $l(T) - q(T)$ share in common having a *maximal full binary tree* subgraph—that is, it is symmetric and every leaf at a given depth branches until the lowest level. This is stated in the following conjecture.

Conjecture 3.7. *For any rooted binary tree* $T = (V, E)$ *with* $|V| = n$, $\mu_{L_T}(1) = l(T) - q(T)$ *unless there is some subgraph* U *of* T *where, given the maximum possible* k *where* $n > 2^k - 1$, U *is a full binary tree of size* $2^k - 1$ *or* $2^{k-1} - 1$. *In this case,* $l(T) - q(T) + 1 \leq \mu_{L_T}(1) \leq l(T) - q(T) + 2$.

On the other hand, with respect to the `XBAR` and `ACC` trees, $l(T) = q(T)$ (every quasipendant vertex branches exactly once), and thus $l(T) - q(T) = 0$. Experimentation has shown that $\mu_{L_T}(1) = 1$ for all $T \in \texttt{XBAR(n)} \cup \texttt{ACC(m)}$, meaning that

$$\mu_{L_T}(1) = l(T) - q(T) + 1$$

for every tree in this class.

So for all three syntactic classes, the lower bound provided by Corollary 3.6 is extremely tight.

We can directly connect this to the syntactic constraints from which we defined these graphs. From the graphical/syntactic perspective, the multiplicity of the eigenvalue 1 in the Laplacian spectrum of these trees indicates an integral part of the syntactic classes' distinction: whether or not the *syntactic constraints* mandate binary branching at quasipendent vertices.

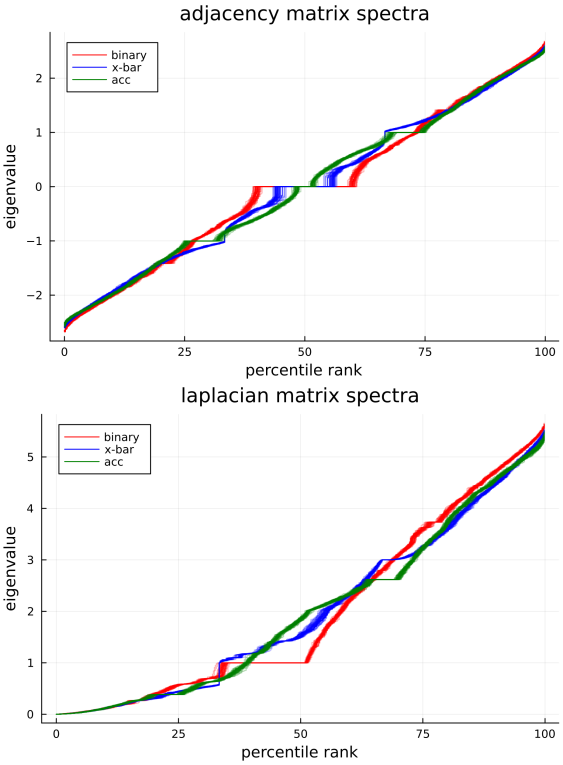


Figure 2: The adjacency and Laplacian spectra of a random sample of 50 trees from each of the three classes `Bin(501)`, `Xbar(501)`, `ACC(170)`.

It is known that eigenvalues with high multiplicity within the spectrum of a graph can indicate the existence of a *motif*, i.e. repeated subgraph, in the graph (Banerjee and Jost, 2009). Recall that Corollary 3.6 linked the multiplicity of eigenvalue 1 in the Laplacian spectra of binary tree graphs to the number of quasipendant vertices branching into two leaves. We can then reframe the discussion around Corollary 3.6 as $\mu_L(1)$ in binary tree graph spectra being potentially indicative of the motif `bin_base` at the leaves of the binary trees.

We now move to discussing the general shape of the eigenvalue graphs and explore potential reasons the spectral graphs preserve class distinctions.

He et al. (2000) observe that the Laplacian spectrum of $k-$ary trees resemble a Cantor step function. The binary branching trees are $3-$ary trees for all non–leaf nodes except the central/root node

[8] Useful discussion provided by Dadedzi (2018).

[9] This is significant given that these trees have over 500 nodes (and subsequently, ove 500 eigenvalues), and yet the multiplicity of eigenvalue 1 is so close to exactly the quantity $l(T) - q(T)$.

branching $2 = k-1$ times, so this substantiates the observation that the Laplacian spectrum of the class `Bin(501)` resembles the Cantor step function.

The Cauchy Interlacing Theorem describes properties of spectrum of submatrices of matrices in relation to the matrix, and can be used to understand properties of the spectrum of subgraphs of graphs as a function of the graph.

Theorem 3.8 (Cauchy Interlacing Theorem, Haemers, 1995)**.** *Let A be an $n \times n$ hermitian matrix (i.e. $A = \overline{A}^T$: it is equal to its conjugate transpose, which is true for any symmetric matrix over the field $\mathbb{R}$) with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, and B be an $m \times m$ submatrix obtained from A by deleting $n-m$ rows and columns of the same index. Suppose B has eigenvalues $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_m$, then*

$$\lambda_i \geq \beta_i \geq \lambda_{n-m+i}, \text{ for } i = \{1, 2, \ldots, m\}.$$

In other words, the eigenvalues of any submatrix of a matrix (where the submatrix is formed by deleting corresponding rows and columns) are interleaved with the eigenvalues of the matrix. Thus, we can generalize this to adjacency matrices of graphs. [10]

Proposition 3.9. *Let $G = (V, E)$ be a graph with $|V| = n$, adjacency matrix A_G and corresponding spectra $\Lambda(A_G) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. Let $H = (V', E')$ be a subgraph of G with $|V'| = m$, adjacency A_H and spectrum $\Lambda(A_H) = \{\mu_1, \mu_2, \ldots, \mu_m\}$. Then*

$$\lambda_i \geq \mu_i \geq \lambda_{n-m+i}, \text{ for } i = \{1, 2, \ldots, m\}.$$

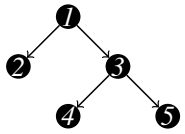
So the eigenvalues of the adjacency matrix of any subgraph of a graph should be interleaved with the eigenvalues of the adjacency matrix of that graph. Recalling that these trees are built off of recursively combining smaller subtrees, this helps give intuition towards the consistent distinctness of the spectra as you increase the size of the tree—given a large tree, the eigenvalues of a subtree of it are distributed amongst the eigenvalues of the tree, preserving the shape, so inductively this is true as you decrease the size of the tree.

[10] Laplacian matrices are more difficult, as the Laplacian of a subgraph of a graph is not immediately a submatrix of the Laplacian of the graph: deleting rows and columns results in a decrease of the degrees reported along the diagonal.

4 Spectral properties of directed syntactic structures

We now consider what happens when we incorporate more traditional assumptions concerning syntactic structure and represent syntactic structures as directed graphs. As above, we explore the spectra of the three classes `BIN`, `XBAR`, and `ACC` as digraphs. Consider the following tree in `Bin(5)` from example 2.4 but with directed edges.

Example 4.1. *The directed rooted binary tree $G = (V, E)$ where $V = \{1, 2, 3, 4, 5\}, E = \{\{1, 2\}, \{1, 3\}, \{3, 4\}, \{3, 5\}\}$:*



First, as in Example 2.4, we calculate the Laplacian of the above digraph.

Example 4.2. *Let $G = (V, E)$ be given as in example 4.1.*[11] *Then*

$$L_G = D_G - A_G = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Observe that both A_G and L_G are *upper triangular* matrices–that is, all the entries below the diagonal are 0. In fact, A_G is *strictly upper-triangular*, as its diagonal too is all 0.[12] We state the following well–known fact about upper triangular matrices.

Proposition 4.3. *Let $M \in \mathbb{R}^n \times \mathbb{R}^n$ be an upper triangular matrix. Then its eigenvalues are the diagonal entries of the matrix.*

The following is derived from Proposition 4.3.[13]

Proposition 4.4. *Let M be an $n \times n$ strictly upper triangular matrix. Then it has one distinct eigenvalue 0 with $\mu_M(0) = n$.*

[11] Note that we say a directed edge $\{v_i, v_j\}$ exists if there is an edge *from v_i to v_j*, and not vice–versa, and recall that the degrees of the nodes here are calculated by using the outdegree.

[12] These graphs are acyclic and thus loopless, so there is never an edge $\{v_i, v_i\}$ from a vertex to itself.

[13] Note that Proposition 4.4 can also be derived by the fact that a strictly upper triangular matrix is *nilpotent*, i.e. for a nilpotent $n \times n$ matrix N there exists a $k \in \mathbb{N}$ such that $N^k = \mathbf{0}$, the $n \times n$ zero matrix. It is a well–known fact that all nilpotent matrices have spectra containing one unique eigenvalue, 0 (with multiplicity equal to the dimension of the matrix).

So we can calculate the eigenvalues of these matrices simply by extracting their diagonal entries. Thus, $\Lambda(A_G) = \{0^5\}$ and $\Lambda(L_G) = \{0^3, 2^2\}$.

This leads us to the following theorem.[14]

Theorem 4.5. *Given a rooted tree digraph $T = (V, E)$ where $|V| = n$, the spectrum of its adjacency matrix A_T is $\{0^n\}$ and the spectrum of its Laplacian matrix L_T is equal to the outdegree of each of its nodes (in particular, $\mu_{L_T}(0) = l(T)$). That is,*

$$\Lambda(A_T) = \{0^n\} \text{ and } \Lambda(L_T) = \mathcal{D}(T).$$

So for any rooted tree digraph, we need only track of the outdegree of each node in order to know the spectrum of its Laplacian. Then we have the following.

Theorem 4.6. *Let $T = (V, E)$ be a directed binary tree with $|V| = n$. Then the spectrum of its Laplacian is $\Lambda(L_T) = \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\}$.*

Next, we state analogous theorems for XBAR and ACC. The proofs are left to the reader—factors to consider are included in the proof of the previous theorem.

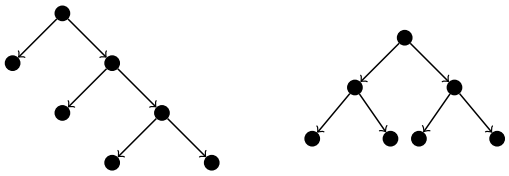
Theorem 4.7. *Let $G = (V, E)$ be an X–bar tree with $|V| = n$. Then the spectrum of its Laplacian is $\Lambda(L_G) = \{0^{\frac{n}{3}}, 1^{\frac{n}{3}+1}, 2^{\frac{n}{3}-1}\}$.*

Theorem 4.8. *Let $T = (V, E) \in \texttt{ACC(m)}$ with $|V| = n$. Then $\Lambda(L_T) = \{0^{m+1}, 1^{n-(2m+1)}, 2^m\}$.*

Given the important role that the spectrum of a graph plays in determining what class it falls in, we might ask the question of whether the spectrum uniquely determines a specific graph G modulo vertex relabeling. For the case of the spectrum of the Laplacian of a directed tree (where the eigenvalues are the degrees) the answer is no, as the following example illustrates.

Example 4.9. *Consider the following graphs.*

Tree T_α: *Tree T_β:*



$T_\alpha, T_\beta \in \texttt{Bin(7)}, \mathcal{D}(T_\alpha) = \mathcal{D}(T_\beta)$, but $T_\alpha \neq T_\beta$.

On the other hand, does the spectra of the Laplacian of these families of graphs, i.e. the outdegrees of the nodes, uniquely determine whether a tree belongs to a specific syntactic class? The answer is *yes* with respect to any family of binary trees—in fact, in general for any n–ary trees (where each node has outdegree of either n or 0).

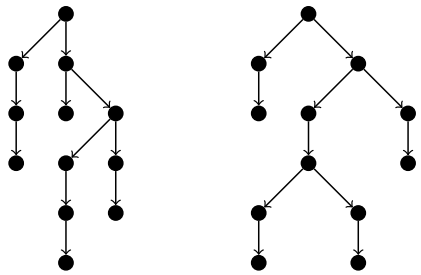
Proposition 4.10. *Let $\mathcal{T}_n$ be a family of $n-$ ary trees, where every non–leaf has an outdegree of n. For total number of nodes N in the tree T, $\Lambda(T) = \{0^{\frac{N+1}{n}}, n^{\frac{N-1}{n}}\}$ if and only if $T \in \mathcal{T}_n$.*

This does not hold for any class non–n–ary trees, i.e. any tree with more than two distinct outdegrees. Whenever more than one non-zero branching factor is allowed, spectral uniqueness is lost.[15]

For instance, XBAR and ACC, two examples of tree families with three distinct eigenvalues/outdegrees (0, 1 and 2), are not uniquely defined by their outdegrees/spectra.

Example 4.11. *Consider the following graphs.*

Tree T_α: *Tree T_β:*



It is clear that $T_\alpha \in \texttt{XBAR(12)}$, $T_\beta \in \texttt{ACC(3)}$, while $T_\beta \notin \texttt{XBAR(12)}$, $T_\alpha \notin \texttt{ACC(3)}$. However, $\Lambda(T_\alpha) = \mathcal{D}(T_\alpha) = \mathcal{D}(T_\beta) = \Lambda(T_\beta)$. So although T_α and T_β are members of distinct syntactic classes, their Laplacian spectra are identical.

Though the spectra of a rooted tree digraph does not definitively classify it to a particular syntactic class (besides n–ary trees), we can say something interesting about spectra of graphs in tree languages generated by (directed) regular tree grammars.

Definition 4.12. *A* regular tree grammar *is a tuple $G = (N, \Sigma, R, S)$. N is a finite set of nonterminals and Σ is a ranked alphabet of terminals such that $\Sigma \cap N =$, $S \in N$ is the initial nonterminal, and R is a finite set of rules of the form $A \to t$ with $A \in N$ and $t \in T_\Sigma(N)$. The tree language generated by G, denoted $L(G)$, is defined as $L(H)$ where H is the context–free grammar $(N, \Sigma \cup \{[,]\}, R, S)$.*

[14] All proofs are contained in the appendix.

[15] Given any tree with at least two nodes with distinct, nonzero branching numbers, you can swap their location (along with the subtrees that they each are the root of) in the tree and come up with a new, distinct tree from the original with the same spectrum.

Assuming that the graphs $t \in T_\Sigma(N)$ comprising the right side of the rules are *directed*, we can state the following theorem.

Theorem 4.13. *Suppose $G = (N, \Sigma, R, S)$ is a (directed) regular tree grammar. Define the set of outdegrees of any rule $A \to t \in R$ for $t \in T_\Sigma(N)$, $\mathcal{OD}(A)$, as the set of outdegrees of the graph structure t excluding any nodes labeled by nonterminals. Then the spectrum of any tree $T \in L(G)$ generated by G is the union of the spectra of the rules used to generate T, i.e. the union of the set of outdegrees of each rule. So, for $\mathcal{R}(T)$ as the set of rules applied to generate T,*

$$\Lambda(L_T) = \bigcup_{R \in \mathcal{R}(T)} \mathcal{OD}(R)$$

In other words, one can directly compute the spectrum of a tree T generated by a directed regular tree grammar by simply taking the union of the outdegrees of the rules used to generate T (excluding any nonterminals, which end up being replaced by nodes of graphs of other rules).

Thus far, our discussion has been focused on the eigenvalues of a matrix representation of a graph. Included in the set of spectral properties of a matrix are its eigenvectors. We now briefly consider the eigenvectors of matrix representations of the syntactic classes we have concerned ourselves with.

With respect to adjacency, Laplacian and signless Laplacian matrix representations, the eigenvectors of all three undirected graph classes all contain both positive and negative signed entries. In comparison, for the directed versions of all of these tree graphs, there is an eigenvector for each eigenvalue whose non–zero entries are all the same sign.[16]

Theorem 4.14. *For any directed rooted tree graph $T = (V, E)$ where $T \in$ `BIN`, `XBAR`, or `ACC`, for every eigenvalue λ of L_T there exists an eigenvector ψ such that every entry of ψ has the same sign.*

5 Conclusion

This paper presents a novel way to explore differences in syntactic structure. We give the first results connecting properties of spectra to syntactically relevant classes of trees. The case study in this paper considers three specific classes of tree structures and shows structural syntactic differences are perceivable at the spectral level, with a variety of properties of these trees (which class they belong to, whether they are directed or undirected, etc.) reflected in the spectra and eigenvectors.

At present, we have only considered a limited set of syntactic classes. This leaves a wide variety of other potentially syntactically relevant graphs, including those that limit leftward branching, or non-tree structure graphs allowing multidominance. Our results leave open further exploration of other classes of trees that are uniquely characterized by the spectra of directed or undirected graphs. We leave this for future work.

One especially exciting result in the current work concerns the degree to which spectra of a class can be derived from a regular tree grammar that generates the class. Just as the Parikh mappings of strings can be derived from the underlying string CFG, so too can the Laplacian spectra of directed syntactic tree graphs be derived from the underlying graph rules. We leave it as an open question to look at richer classes of tree grammars and alternative matrix representations.

Our motivation in this work is to identify novel mathematical tools with which we can look beneath surface representations of linguistic structures and explore more fundamental features of their linguistic essence. The current work has utilized spectral graph theory as one mathematical tool to do just this, examining the reflection of certain syntactic features and properties in the spectra. This paper demonstrates SGT is a way to peel back the surface combinatorial graphical structure we see, and attempt to understand deeper, more inherent features of the syntactic structures. The goal of future work would be to take this one step further—not only understanding the ways in which spectra can reflect syntactically relevant properties, but further developing the spectral studies of these graphs in order to use the spectra to identify fundamental properties about syntactic structure that are inaccessible or hidden from view based on the surface combinatorial structure of these graphs.

Acknowledgements

The authors would like to thank Daniel Spielman for his invaluable guidance and insights.

[16] As eigenvectors define a linear space, each eigenvector defines a set of all multiples of that eigenvector by all real numbers. So this is equivalent to saying an eigenvector's entries do not change signs.

References

Ali Zeydi Abdian, Afshin Behmaram, and Gholam Hossein Fath-Tabar. 2018. Graphs determined by sign-

less Laplacian spectra. *AKCE International Journal of Graphs and Combinatorics*.

Anirban Banerjee and Jürgen Jost. 2009. Graph spectra as a systematic tool in computational biology. *Discrete Applied Mathematics*, 157(10):2425–2431.

J. A. Bondy and U. S. R. Murty. 1976. *Graph Theory with Applications*. Elsevier, New York.

Koel Chowdhury, Cristina España-Bonet, and Josef Genabith. 2021. Tracing Source Language Interference in Translation with Graph-Isomorphism Measures. In *Proceedings of Recent Advances in Natural Language Processing*, pages 375–385.

Fan Chung. 1997. *Spectral Graph Theory*. American Mathematical Society.

Dragos M Cvetkovic, Michael Doob, and Horst Sachs. 1980. *Spectra of Graphs : Theory and Application*. Academic Press.

Kenneth Dadedzi. 2018. *Analysis of tree spectra*. Ph.D. thesis, Stellenbosch University.

Robert Frank and Fero Kuminiak. 2000. Primitive Asymmetric C-Command Derives X-Theory. In *Proceedings of the 30th Annual Meeting of the North East Linguistics Society*, volume 1, pages 203–217, University of Massachusetts, Amherst.

Robert Frank and K. Vijay-Shanker. 2001. Primitive C-Command. *Syntax*, 4:164–204.

Chris Godsil and Gordon Royle. 2001. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer.

Willem H. Haemers. 1995. Interlacing eigenvalues and graphs. *Linear Algebra and its Applications*, 226–228:593–616.

Willem H. Haemers and Edward Spence. 2004. Enumeration of cospectral graphs. *European Journal of Combinatorics*, 25(2):199–211.

Li He, Xiangwei Liu, and Gilbert Strang. 2000. Laplacian Eigenvalues of Growing Trees. In *Conf. on Math. Theory of Networks and Systems*.

A. J. Hoffman. 1970. On Eigenvalues and Colorings of Graphs. In *Graph Theory and Its Applications*, volume 175, pages 79–92. Academic Press, New York.

David P. Jacobs, Elismar R. Oliveira, and Vilmar Trevisan. 2021. Most Laplacian eigenvalues of a tree are small. *Journal of Combinatorial Theory, Series B*, 146:1–33.

Richard Kayne. 1994. *The Antisymmetry of Syntax*. MIT Press, Cambridge, MA.

Fero Kuminiak. 1999. Formal properties of asymmetric c-command in tree structures. Manuscript, Johns Hopkins University, Baltimore, MD.

Brendan McKay. 1977. On the spectral characterisation of trees. *Ars Combin*, 3:219–232.

Eva Nosal. 1970. Eigenvalues of graphs. Master's thesis, University of Calgary.

Andrew Ortegaray, Robert Berwick, and Matilde Marcolli. 2021. Heat Kernel Analysis of Syntactic Structures. *Mathematics in Computer Science*, 15.

John H Smith. 1970. Some properties of the spectrum of a graph. *Combinatorial Structures and their applications*, pages 403–406.

Daniel A. Spielman. 2019. Spectral and Algebraic Graph Theory. Manuscript.

Edwin R. van Dam and Willem H. Haemers. 2003. Which graphs are determined by their spectrum? *Linear Algebra and its Applications*, 373:241–272.

Wei Wang and Cheng-Xian Xu. 2006. On the spectral characterization of T-shape trees. *Linear Algebra and its Applications*, 414(2):492–501.

Herbert S. Wilf. 1967. The Eigenvalues of a Graph and Its Chromatic Number. *Journal of the London Mathematical Society*, s1-42(1):330–332.

A Appendix: Proofs of Theorems

Theorem 4.5. *Given a rooted tree digraph $T = (V, E)$ where $|V| = n$, the spectrum of its adjacency matrix A_T is $\{0^n\}$ and the spectrum of its Laplacian matrix L_T is equal to the outdegree of each of its nodes (in particular, $\mu_{L_T}(0) = l(T)$). That is,*

$$\Lambda(A_T) = \{0^n\} \text{ and } \Lambda(L_T) = \mathcal{D}(T).$$

Proof of Theorem 4.5. Suppose $T = (V, E)$ is a rooted tree digraph with $|V| = n$. There exists an enumeration of the vertices such that $i = e(v_i) \leq e(v_j) = j$ for natural numbers i, j iff v_i is the parent of v_j. Then, for all directed edges $\{v_i, v_j\}, i \leq j$. As (i, j) corresponds to the indices of the adjacency matrix A_T of G, this yields a strictly upper-triangular adjacency matrix A_T: all edges from i to j will set $a_{ij} = 1$, above the diagonal, and $a_{ji} = 0$, below the diagonal.[17]

Given that the adjacency matrix is strictly upper triangular, its spectrum is

$$\Lambda(A_T) = \{0^n\}$$

.

The diagonal entries of the Laplacian are determined by D_G, which correspond to the outdegree

[17] As there are no self-loops in G, meaning no edges $\{v_i, v_i\}$ for all $i \leq n$, $a_{ii} = 0$ for all $i \leq n$.

of each of the n nodes. L_G is upper triangular: $L_G = D_G - A_G$.[18] Thus, by Proposition 4.3, the eigenvalues of L_G are equal to the diagonal D_G, which is the outdegree of each of the n nodes.

□

Theorem 4.6. *Let $T = (V, E)$ be a directed binary tree with $|V| = n$. Then the spectrum of its Laplacian is $\Lambda(L_T) = \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\}$.*

Proof of Theorem 4.6. We can think about a given binary tree as a construction starting with the smallest possible binary tree, the 3–noded binary tree, and then recursively substituting that same binary tree with 3 vertices to the leaves of the first tree. Any binary tree with $n = 2k + 1$ vertices can be constructed by inserting $(n - 3)/2$ copies of this base binary tree, root–to–leaf (i.e. the root of the tree being inserted inserts into one of the current leaves) including the initial starting tree, or $(n - 1)/2$ copies of the base binary tree total.

We prove this by induction. For $k = 1$ with $n = 3$ we have $T_3 =$ `bin_base`, which has one outdegree of 2 (the root/branching node) and two outdegrees of 0 (the leaves). So $\Lambda(T_3) = \{0^2, 2^2\} = \{0^{\frac{3+1}{2}}, 2^{\frac{3-1}{2}}\} = \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\}$.

Suppose we have performed $k-2$ insertions into this tree T_{2k-1} (for a total of $k - 1$ copies of the binary tree). At each insertion of a new base binary tree T_3 to one of the leaves of the current binary tree, two additional nodes are gained. The first tree T_3 begins with 3 nodes, and each subsequent insertion of a new copy of T_3 yields two more nodes (inserting root-to-leaf does not add a count to the node with the root node, but it does with the two new leaves). So $n = |V| = 2(k - 1) + 1 = 2k - 1$. Assume $\Lambda(T_{2k-1}) = \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\} = \{0^{\frac{2k-1+1}{2}}, 2^{\frac{2k-1-1}{2}}\} = \{0^k, 2^{k-1}\}$.

To construct T_k, we insert a new copy of the base tree T_3 to one of the leaves of T_{2k-1}. This insertion forces that leaf to branch, turning its outdegree from 0 to 2, and then adds two new outdegrees of 0, the two new leaves, resulting in a net gain of one leaf. We have gained one node with outdegree 2, the formerly-leaf-turned-binary-branch. Thus, the insertion of a copy of T_3 into T_{2k-1} has a net degree gain of one 2-degree and one 0-degree. Note the total number of nodes here is two more than $2k - 1, 2k + 1$. Thus

$$\Lambda(T_{2k+1}) = \{0^{k+1}, 2^{k-1+1}\} = \{0^{k+1}, 2^k\}$$

$$= \{0^{\frac{2k+2}{2}}, 2^{\frac{2k}{2}}\} = \{0^{\frac{(2k+1)+1}{2}}, 2^{\frac{(2k+1)-1}{2}}\}$$

$$= \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\}.$$

The proof of the class `XBAR` is identical in structure: we only need observe that substituting `xbar` to the specifier or complementizer positions adds three nodes to the graph (as we create a new edge) and increases the node counts by one new outdegree 2, one outdegree of 1 and one outdegree of 0.

To prove the case of `ACC`, we are forced to consider the multiplicity of eigenvalues as a function of both the number of insertions and the number of nodes. This is due to the variability in the number of nodes gained through each different operation. Operation (1) above creating the five–noded structure results in a net gain of one outdegree 0, two outdegrees of 1, and one outdegree of 2. Operation (2), replacing the quasipendant node and its leaf with the four– or five–noded structure results in a net gain of one outdegree 0, one outdegree of 1, and one outdegree of 2 or one outdegree 0 and one outdegree 2. This optionality of which structure you insert, as well as the ambiguity of indexing this class by number of insertions as opposed to node number (for this very reason) results in the variation of multiplicity of eigenvalue 1.

□

Proposition 4.10. *Let $\mathcal{T}_n$ be a family of $n-$ary trees, where every non–leaf has an outdegree of n. For total number of nodes N in the tree T, $\Lambda(T) = \{0^{\frac{N+1}{n}}, n^{\frac{N-1}{n}}\}$ if and only if $T \in \mathcal{T}_n$.*

Proof of Proposition 4.10. Given an $n-$ary rooted directed tree $T = (V, E)$ with $|V| = n$, any non–leaf branches exactly n times by definition. So every node either has n children or is a leaf. Thus by Theorem 4.5, $\Lambda(\mathcal{T}_n) = \{0^{\frac{N+1}{n}}, n^{\frac{N-1}{n}}\}$.

On the other hand, suppose we are given a rooted tree digraph $T = (V, E)$ with spectrum $\Lambda(T) = \{0^{\frac{N+1}{n}}, n^{\frac{N-1}{n}}\}$ for N nodes and $n \in \mathbb{N}$. Since any rooted tree digraph has eigenvalues corresponding directly to its outdegrees means (by Theorem 4.5) T must have $\frac{N+1}{n}$ leaves and $\frac{N-1}{n}$ nodes with outdegree n. Thus $T \in \mathcal{T}_n$. □

[18] Both D_G and A_G are upper triangular, and the sum/difference of two upper triangular matrices is upper triangular.

Theorem 4.14. *For any directed rooted tree graph $T = (V, E)$ where $T \in$ BIN, XBAR, or ACC, for every eigenvalue λ of L_T there exists an eigenvector ψ such that every entry of ψ has the same sign.*

Proof of Theorem 4.14. Recall that an eigenvector of any matrix is, by definition 2.6, nonzero. We provide the intuition behind the class BIN, as the other two are similar.

Let $T = (V, E)$ where $T \in \texttt{BIN(n)}$ is a directed rooted tree graph where $n = 2k + 1$ for some $k \in \mathbb{N}$. We know the Laplacian L_T is upper triangular. It will have $k + 1$ rows/columns of zeros, corresponding to each of the $k + 1$ leaves.As each binary tree with $n = 2k + 1$ nodes has k binary–branching nodes, L_T has k rows/columns with 2 on the diagonal (i, i) and two entries of -1, at (i, j_1) and (i, j_2), where $j_1, j_2 > i$. For any (nonzero) eigenvector

$$\psi = [\psi_1, \psi_2, \ldots, \psi_n]^T$$

where

$$L_T\psi = \lambda\psi,$$

the zero rows of L_T indexed by integers $l_1, l_2, \ldots, l_{k+1}$ give rise to $k + 1$ equations of the form

$$0 = \lambda\psi_{l_i}.$$

On the other hand, the k nonzero rows give rise to equations of the form

$$2\psi_i - \psi_{i+c} - \psi_{i+c+d} = \lambda\psi_i$$

for nonzero numbers $c, d \in \mathbb{N}$.

It is useful in building intuition to connect the occurrences of each $\psi_i \in \psi$ in the system of equations given by the equation

$$L_T\psi = \lambda\psi$$

to the behavior of the node in the graph enumerated with label i.

Let $\mathcal{L}$ be the set of integers corresponding to the labels of the leaves of the tree. For all $l \in \mathcal{L}$, ψ_l exists as a variable with coefficient -1 in exactly one equation of the second form, that is,

$$2\psi_i - \psi_j - \psi_l = \lambda\psi_i,$$

as every leaf node necessarily is connected to one binary–branching node, as well as in one equation of the first form,

$$0 = \lambda\psi_l.$$

Every non–leaf, non–root node with label m exists in two equations, both of the second form: one with coefficient 2, that is,

$$2\psi_m - \psi_i - \psi_j = \lambda\psi_m,$$

and one with the coefficient -1,

$$2\psi_i - \psi_j - \psi_m = \lambda\psi_i.$$

The root node r exists in exactly one equation, the equation of the second form, with coefficient 2:

$$2\psi_r - \psi_i - \psi_j = \lambda\psi_r.$$

In what follows, we assume the matrix has been permuted into the form of the first k rows being the nonzero rows, that is, 2 in the diagonal followed later in the row with two entries of -1 (i.e. the rows corresponding to the binary–branching nodes) and then $k + 1$ rows of zeros. Schematically, the matrix is of the form:

$$A = \begin{bmatrix} 2 & a_{12} & & \cdots & & a_{1n} \\ 0 & \ddots & & \cdots & & \vdots \\ & & 2 & a_{k,(k+1)} & \cdots & a_{k,n} \\ \vdots & & \ddots & 0 & \cdots & 0 \\ & & & & \ddots & \vdots \\ 0 & & \cdots & & 0 & 0 \end{bmatrix}$$

A is not only an upper triangular matrix, but also the last $k + 1$ rows is an all–zeros rectangle of dimension $(k + 1) \times n$.

There are two categories of eigenvectors, those which pertain to eigenvalue 2 and those pertaining to eigenvalue 0.

Case 1: Suppose $\lambda = 2$.

Then there are $k + 1$ equations of the form

$$0 = 2\psi_{l_i},$$

yielding

$$\psi_{l_i} = 0.$$

By the form of the vector above, the final nonzero row of the matrix L_T, row k, with a 2 in position (k, k), gives the equation

$$2\psi_k - \psi_{j_1} - \psi_{j_2} = 2\psi_k$$

will subsequently have

$$\psi_{j_1} = \psi_{j_2} = 0.$$

Intuitively, we can understand this row as representing a binary–branching node in the tree which branches into two non–branching leaf nodes. There is, necessarily, at least one of these existing in any given tree. Then this results in

$$2\psi_k - 0 - 0 = 2\psi_k,$$

yielding ψ_k being a free variable (where k is not the label of the root node, assuming $k > 1$, that is, $2k + 1 > 3$).

It is necessary for $\psi_k = 0$, and subsequently for $\psi_j = 0$, in the second equation containing ψ_k,

$$2\psi_h - \psi_k - \psi_j = 2\psi_h.$$

For the leaf nodes, then, it is easy to see that the fact that for every $l \in \mathcal{L}$, $\psi_l = 0$ results in free variables for the k binary–branching nodes, which all exist in a second equation with coefficient -1, except for the root node. The reader can verify that then for every ψ_i in at least two equations, that is, every entry of the eigenvector except for the first (which correlates to the root node and is only in equations of the second form with coefficient 2), $\psi_i = 0$. As eigenvectors must be nonzero, then, this first entry must assume a nonzero value. So every eigenvector of eigenvalue 2 must have eigenvector of the form $c \cdot e_1$ for the first basis vector e_1 and $c \in \mathbb{R}$. As 2 has multiplicity k, there are k eigenvectors of this form.

Case 2: Suppose $\lambda = 0$.

Then equations of the first form are

$$0 = 0\psi_l$$

and the second form are

$$2\psi_k - \psi_{j_1} - \psi_{j_2} = 0.$$

This means that every $l \in \mathcal{L}$, ψ_l becomes a free variable. The reader can verify that in order for a given eigenvector to have all entries of either the same sign or 0, exactly one ψ_l can be nonzero. Not only this, but for $\psi_1 = c$ for $c \in \mathbb{R}$ and root with label 1, for each node i on the path from root to leaf l with nonzero ψ_l,

$$\psi_i = 2^m c$$

for m being the length of the path from root to i. This comes from the equations of the second form

$$2\psi_k - \psi_{j_1} - \psi_{j_2} = 0$$

as, without loss of generality, if $j_1, j_2 \in \mathcal{L}$ and $\psi_{j_1} = 0$,[19] then

$$2\psi_k = \psi_{j_2}.$$

Each nonzero entry of a given eigenvector thus corresponds to the labels of nodes which form a directed path from root to leaf for a chosen nonzero ψ_l corresponding to label of a leaf l.

Therefore, each of the $k + 1$ eigenvectors of eigenvalue 0 correspond to each possible nonzero choice of ψ_l for $l \in \mathcal{L}$, and each of these eigenvectors have nonzero entries ψ_i for every label i on the path from the root to l for the given nonzero ψ_l.

In the case where $T \in \texttt{XBAR(n)}$ or $T \in \texttt{ACC(k)}$, note that we have the additional row/column case where there is 1 in the diagonal and thus nonzero, non–two rows are of the form $\psi_i - \psi_{i+c} = 1\psi_i$, yielding ψ_i as a free variable and $\psi_{i+c} = 0$. □

[19] The case where both $\psi_{j_1} = \psi_{j_2} = 0$ results in $\psi_k = 0$, which percolates into the equation where ψ_k has coefficient -1 and the same scenario is repeated.

Parasitic gaps in Japanese: An MG-based approach

Yu Tomita
Institut für Linguistik
Universität Leipzig
ytomita@uni-leipzig.de

Hitomi Hirayama
Keio University
hhirayam@keio.jp

Abstract

This paper aims to provide an account based on Minimalist Grammar (MG) for what are called parasitic gaps in Japanese, which we take as a null pronoun. The main goal of this paper is to provide a syntactic account of the environment in which a parasitic gap reading is licensed in Japanese. First, Japanese parasitic gaps are compared with English ones, illustrating the puzzle to be solved. We argue that the possibility of co-indexing between a parasitic gap (null pronoun) and wh-phrase is correlated with the point at which the wh-phrase enters the derivation. We also show that scrambling counterbleeds licensing of parasitic gaps by using extended Directional Minimalist Grammar. The proposed syntactic analysis has an advantage over a semantic analysis in that there is no need to postulate vacuous movement of the subject wh-phrase.

1 Introduction

Minimalist Grammar (MG, Stabler, 1997a,b) has been mainly applied to Indo-European languages (especially English), although it was inspired by Chomsky's influential work (Chomsky, 1995), which has been cited in "minimalist" syntax works on various languages. In this paper, we attempt to apply MG to explain a phenomenon in Japanese that is substantially different from that in English. This phenomenon is a parasitic gap construction, which has been vigorously explored in both the Japanese and English syntax literature, including one based on MGs. Adopting the idea that Japanese parasitic gaps are null pronouns, we propose that the c-command relation in the derivation is key to explaining the confounding behavior in the co-indexed reading between a *pro* and wh-phrase in Japanese.

The remainder of this paper is structured as follows: Section 2 provides a basic background on the phenomenon of interest. Section 3 introduces the basic tools used in the proposed analysis, i.e., Directional Minimalist Grammar with some extensions. Section 4 proposes our generalization on how the derivation accounts for a possible co-index configuration. We discuss the proposed analysis and compare it with formalism in other MG literature and with other works on Japanese and English parasitic gaps in Section 5. Section 6 concludes the article with implications for future research.

2 Background: "Parasitic gaps" in Japanese

This section provides a general background on parasitic gaps in Japanese. There are several approaches to Japanese parasitic gaps, but in this paper, we take a null pronominal account as a starting point. This assumption already suggests that the parasitic gaps in Japanese are radically different from those in English. Nevertheless, there is one common feature between the two languages, namely, obligatory movement, and we introduce the puzzle of co-indexation. A short introduction of other characteristics of Japanese parasitic gaps follows before providing a formal tool.

2.1 Nature of parasitic gaps in Japanese and the co-indexation problem

A parasitic gap *pg* is defined as a gap that requires another gap to be grammatical. A typical example in English from Engdahl (1983, 5) is shown in (1). The parasitic gap is inside a syntactic island indicated by square brackets.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 212-221.
Amherst, Massachusetts, June 15-17, 2023

(1) Which article$_i$ did John file t_i [without reading pg_i]?

Several languages are reported to have parasitic gaps. Japanese is one of them, and there have been debates over the nature of parasitic gaps in Japanese (Abe, 2011; Takahashi, 2006; Yoshimura, 1992). When there is a gap in Japanese, we have multiple candidates: a trace of null operator movement, the result of ellipsis, or a null pronoun *pro*. In this paper, we take the last approach by Hirayama (2018), which follows Yoshimura (1992).

An example sentence with a parasitic gap in Japanese is shown in (2), which has a gap inside the subject island. The sentence involves a movement of *wh*-phrase *dare-o*. This interrogative sentence can be used to identify a poor man criticized by a person they met for the first time.

(2) Dare$_i$-o [hazimete pg_i atta
who-ACC for the first time saw
hito]-ga t_i kenasimasitata ka?
person-NOM criticized Q
'Who was it that a person who saw pg for the first time criticized t?'

Note that (2) also has a reading where a parasitic gap and *dare* 'who' refer to different individuals. In this case, a parasitic gap refers to a contextually salient entity. Throughout this paper, our focus is on whether a parasitic gap inside an island and the wh-phrase can refer to the same entity. In other words, we explore the environment where a wh-phrase and parasitic gap may be co-indexed.

As we will see in detail later, there are numerous differences between parasitic gaps in Japanese and English, as pointed out by Hirayama (2018). These distinct characteristics suggest that Japanese parasitic gaps are completely different from English ones, and Japanese parasitic gaps should not even be named as such. However, there is one striking similarity; parasitic gaps are licensed by overt movement of the wh-phrase in Japanese as well as in English. The sentence (3a) (Engdahl, 1983, 14) is ungrammatical under the interpretation where the parasitic gap and wh-phrase refer to the same entity, and this is due to the wh-phrase *which article* staying in-situ. The ungrammaticality under the co-indexed reading is obtained in the Japanese example (3b), where the wh-phrase *dare-o* 'who-ACC' stays in-situ. Note that the representative example we saw in (2) is derived from (3b) by moving the wh-phrase from the base-generated position to the sentence-initial position.

(3) a. * I forget who filed which articles$_i$ without reading pg_i

b. * [Hazimete pg_i atta
for the first time saw
hito]-ga dare$_i$-o
person-NOM who-ACC
kenasimasita ka?
criticized Q
(Intended:= (2)

The Japanese example poses a question. In general, Japanese wh-phrases can stay in situ but can scope over syntactic islands except for wh-islands (Shimoyama, 2006), as shown by (4). In the example, even though the wh-phrase is in the adjunct island, the whole sentence can be interpreted as a matrix question.

(4) Taroo-wa [Hanako-ga nani-o
Taro-TOP Hanako-NOM what-ACC
tabeta kara] okotta no?
ate because got angry Q
'For which x did Taro get mad because Hanako ate x?

Furthermore, Japanese null pronouns in a syntactic island can be co-indexed with a DP in a matrix clause without movement, as shown in (5). Note that in the English translation, an overt pronoun is obligatory to obtain the intended reading.

(5) Taroo-wa [*pro*$_i$ tabe-zuni]
Taro-TOP eating-without
keeki$_i$-o suteta.
cake-ACC threw away
'Taro threw away the cake$_i$ without eating it$_i$/*$\emptyset_i$.'

Here is the puzzle: Why is a null pronoun inside the island unable to be co-indexed with the in-situ wh-phrase in (3b)? The semantics of questions allows the wh-phrase to scope over the island. Furthermore, no movement is necessary for a DP to bind a pronoun in (5). Hirayama (2018) gave an answer to this question based on the semantics of questions in Japanese, but in this paper, we try to give an answer from a syntactic perspective.

2.2 Other properties of parasitic gaps in Japanese

Hirayama (2018) summarized the difference between the parasitic gaps in Japanese and English, as shown in Table 1. Among them, the first three differences between English and Japanese are important to account for covariation readings of Japanese parasitic gaps in this paper. They altogether indicate that the configurational requirement of co-indexation of a parasitic gap and the wh-phrase is looser than the environment where English parasitic gaps are licensed; the co-indexed reading is obtained as long as the wh-phrase c-commands the parasitic gap in the surface order.

First, only A'-movement can license English parasitic gaps, as shown by the ungrammaticality of the passive sentence in (6) (Engdahl, 1983, 13). By contrast, (2) involves clause-internal scrambling, which can be A-movement (Saito, 1992), and the co-indexed reading is available.

(6) * John$_i$ was killed t_i [by a tree falling on pg_i].

Next, we have seen that in-situ wh-phrases can never license parasitic gaps in English. As shown in the last section, this is the same in Japanese in most cases. However, when the subject is the wh-phrase, no movement is necessary to obtain the co-indexed reading, as shown in (7) (Hirayama, 2018, 7). Furthermore, (7) also indicates that the anti-c-command condition does not hold in Japanese. The anti-c-command condition states that a real trace cannot c-command a parasitic gap. In other words, the English translation in (7) is ungrammatical.

(7) Dono gakusee-ga Hanako-ni
which student-NOM Hanako-by
[Taroo-ga pg_i sagasu mae-ni]
Taroo-NOM look for before
mitukatta no?
found Q
'Which student$_i$ t_i got found by Hanako before Taro looked for pg_i?'

To summarize, the co-indexed reading in Japanese can be licensed as long as the wh-phrase c-commands *pro*. The type of movement does not matter. The anti-c-command condition does not apply in Japanese, and consequently, it is possible for the subject to be the wh-phrase, and *pro* may be co-indexed with it. Next, we introduce the formalism used in our analysis to account for the characteristics of Japanese parasitic gaps.

3 Directional Minimalist Grammar

A *Minimalist Grammar* (MG, Stabler, 1997a,b) is a mathematically rigorous lexicalized grammar formalism suitable for implementing modern syntactic theory in the (early) *Minimalist Program* (Chomsky, 1995).

An MG contains a set of *lexical items*, each carrying a list of features. For example, a transitive verb *praised*$_{\texttt{=D,=D,V}}$ carries a list of features $\texttt{=D}, \texttt{=D}, \texttt{V}$, where $\texttt{=D}$ is a *selector* of some DP and $\texttt{V}$ a *category*. Intuitively, each structure-building operation is driven by a feature; **merge** saturates a category $\texttt{b}$ with the corresponding selector $\texttt{=b}$, combining two expressions (lexical or phrasal) and building a new phrasal expression.

Some variants of MG assume **adjoin** and **scramble** (Frey and Gärtner, 2002), which allow us to perform the adjunction and scrambling operations on that MG. Kobele (2010) also introduces operations called **assume** and **discharge**.

	Eng	Jpn
Must the antecedent be in an A'-position?	Yes	Could be A-position
Can in-situ wh-phrases license pgs?	No	Subject wh does not need movement
Does the anti-c-command condition hold?	Yes	No (the wh must c-command a pg)
Is a pg island sensitive?	Yes	No
Is there Case-matching Effect?	-	No
What category can be a pg?	NP	NP and PP

Table 1: Characteristics of Japanese parasitic gap constructions (adapted from Hirayama, 2018)

3.1 Syntactic object

We assume that every syntactic object is a pair $\langle \mathrm{A}, \phi \rangle$. A represents a lexeme or binary branching phrasal tree $[\ \Gamma\ \Delta\ ]$, where Γ and Δ are left and right subtrees (= syntactic objects), respectively, and ϕ is a label, namely an unsaturated feature bundle. We will write them as A_{ϕ}. Let us denote by $\mathrm{A} = \langle \mathrm{A}, \emptyset \rangle$ a syntactic object that no longer moves in the course of the derivation. Let us write $\mathrm{A}\langle\Gamma\rangle_{\gamma}$ as a syntactic object that contains an occurrence of a syntactic object Γ, where $\mathrm{A}\langle\_\rangle_{\gamma}$ is called a *syntactic context*, an object equivalent to the syntactic object $\mathrm{A}\langle\Gamma\rangle_{\gamma}$ except an empty placeholder _ which replaces exactly one occurrence of Γ. This definition is extended later in the paper.

3.2 Merge

The standard MG only allows the head-initial phrase, according to Kayne (1994). However, the order of Japanese words appears to be head-final. Therefore, the domain of **merge** contains a lexical item that can select its complement on the left side. In other words, each word specifies a linear order in the result of **merge** (Stabler, 2011).

(8) a. $\textbf{merge}\,(\mathrm{A}_{\mathtt{X},\gamma}, \mathrm{B}_{<\mathtt{X},\phi}) = [\ \mathrm{A}_{\gamma}\ \mathrm{B}\ ]_{\phi}$

b. $\textbf{merge}\,(\mathrm{A}_{\mathtt{X},\gamma}, \mathrm{B}_{>\mathtt{X},\phi}) = [\ \mathrm{B}\ \mathrm{A}_{\gamma}\ ]_{\phi}$

In (8), we give the general definition of **merge** in the DMG. $\mathrm{A}_{\mathtt{X},\gamma}$ has the leftmost category $\mathtt{X}$, while $\mathrm{B}_{<\mathtt{X},\phi}$ has the leftmost selector $<\mathtt{X}$ in (8a) and $\mathrm{B}_{>\mathtt{X},\phi}$ has the leftmost selector $>\mathtt{X}$ in (8b). They comprise a new syntactic object labeled ϕ through **merge**, saturating the leftmost selector feature with a corresponding category feature. Because Japanese word order is supposed to be strongly head-final, we mainly use the rule (8a).

3.3 Move

The MG also has an operation called **move**, which cashes out the displacement. This operation is driven by some (*move*) *licensor feature* $+\mathtt{y}$ and the corresponding *licensee feature* $-\mathtt{y}$.

(9) $\textbf{move}\,(\mathrm{A}\langle\mathrm{B}_{-\mathtt{y},\delta}\rangle_{+\mathtt{y},\phi}) = [\ \mathrm{B}_{\delta}\ \mathrm{A}\langle\epsilon\rangle\ ]_{\phi}$

In (9), a syntactic object A carries the leftmost licensor $+\mathtt{y}$, and a subtree B carries the corresponding leftmost licensee $-\mathtt{y}$. Then, B moves to the specifier position of A, saturating A's $+\mathtt{y}$ feature with B's $-\mathtt{y}$ feature and leaving a phonologically empty element ϵ. If δ is not empty, B continues to move.

Covert movement in MG is similar to *feature movement* and is defined below (10). In this paper, a designated licensee $-\mathtt{q}$ always denotes a covert movement feature.

(10) $\textbf{move}\,(\mathrm{A}\langle\mathrm{B}_{-\mathtt{q},\delta}\rangle_{+\mathtt{q},\phi}) = [\ \epsilon_{\delta}\ \mathrm{A}\langle\mathrm{B}\rangle\ ]_{\phi}$

3.4 First extension: Adjoin and Scrambling

In addition to **merge** and **move**, here we assume two operations called **adjoin** and **scramble**, which are introduced by Frey and Gärtner (2002). Like **merge** and **move**, these are binary and unary operations invoked by different features. In **adjoin** an *adjoin licensor* $\gg\mathtt{X}$ selects a category $\mathtt{X}$ but does not saturate it.

(11) $\textbf{adjoin}\,(\mathrm{A}_{\mathtt{X},\delta}, \mathrm{B}_{\gg\mathtt{X},\eta}) = [\ \mathrm{B}_{\eta}\ \mathrm{A}\ ]_{\mathtt{X},\delta}$

scramble is invoked by a feature called *scramble licensee* $\tilde{}\mathtt{X}$, which behaves like

an adjoin licensor, except that deletion of a scramble licensee is optional.

(12) **scramble** $(\mathrm{A}\langle \mathrm{B}_{\sim \mathtt{X},\eta}\rangle_{\mathtt{X},\delta}) = [\ \mathrm{B}_{(\sim \mathtt{X})\eta}\ \mathrm{A}\langle\epsilon\rangle\ ]_{\mathtt{X},\delta}$

3.5 Second extension: Slash-Feature Percolation

We adopt additional operations proposed by Kobele (2010) as *Slash-Feature Percolation*. In this approach, in addition to **merge-move** and **adjoin-scramble**, we assume two further operations: **assume** and **discharge**. First, a unary operation **assume** takes a syntactic object with a selector $>\mathtt{X}$ and adds a new moving syntactic object called *assumption*, which is a 'dummy lexeme' $[\mathtt{X}, \delta]$ carrying a sequence of (move and scramble) licensees δ.

(13) **assume** $(\mathrm{A}_{<\mathtt{X},\phi}) = [\ [\mathtt{X}, \delta]_{\delta}\ \mathrm{A}\ ]_{\phi}$

The syntactic object containing some assumption coming from (13) can be regarded as some syntactic context $\Gamma\langle\_\rangle$ whose gap is occupied with that assumption. Note that if some syntactic context $\mathrm{A}\langle[\gamma]_{\eqsim \mathtt{Y},\delta}\rangle_{\phi}$ that contains some assumption whose leftmost feature is $\eqsim\mathtt{Y}$ (a move or scramble licensee) undergoes **move** or **scramble**, then the licensee $\eqsim\mathtt{Y}$ in $\eqsim\mathtt{Y}, \delta$ is consumed but $\eqsim\mathtt{Y}$ in the dummy lexeme $[\gamma]$ remains.

The other binary operation **discharge** takes a syntactic object with *assumption* and a corresponding object.

(14) **discharge** $(\Gamma\langle[\gamma, -\mathtt{Y}]_{-\mathtt{Y}}\rangle, \mathrm{B}_{\gamma,-\mathtt{Y},\delta}) = \Gamma\langle\mathrm{B}_{-\mathtt{Y},\delta}\rangle$

This operation replaces an assumption in some syntactic context with some syntactic object. However, we must modify this rule because (i) Kobele (2010, 2012) did not introduce scrambling features and, (ii) as Kobele (2012) wrote, the **assume-discharge** framework can cause an explosion of ambiguous derivations for a single sentence. To avoid these problems, we propose that **discharge** must be applied if and only if a syntactic object contains a gap filled with an assumption that has just deleted some move or scrambling licensee via movement and only carries a single move or scramble licensee $[\gamma, \eqsim\mathtt{Y}]_{\eqsim\mathtt{Y}}$, where $\eqsim\mathtt{Y}$ stands for a move or scrambling licensee. That is, **discharge** only targets some syntactic context with an assumption that moves to the left edge of the tree.[1]

(15) **discharge** $([\ [\gamma, \eqsim\mathtt{Y}]_{\eqsim\mathtt{Y}}\ \mathrm{A}\ ]_{\phi}, \mathrm{B}_{\gamma,\eqsim\mathtt{Y},\delta}) = [\ \mathrm{B}_{\eqsim\mathtt{Y},\delta}\ \mathrm{A}\ ]_{\phi}$

Definition 3.1. A *Directional Minimalist Grammar with Adjunction, Scrambling, assume, and discharge* $G_{\mathcal{P}}$ is a tuple $(\Sigma, B, F, \Lambda, c, \mathcal{P})$, where Σ is a set of (possibly phonetically empty) *words*; B a finite set of *category features*; F a finite set of *licensing features*; Λ a finite set of *lexicon*, whose element, a *lexical entry*, is a pair of a lexeme; and a sequence of features $\phi \in (B_{=} \cup F_{+})^{*}\, B \cup B_{\approx}\, (F_{-} \cup B_{\sim})^{*}$, where $B_{=} = \{>\mathtt{b}, <\mathtt{b} \mid \mathtt{b} \in B\}$ is a finite set of *selection features*, $B_{\approx} = \{\gg\mathtt{b}, \ll\mathtt{b} \mid \mathtt{b} \in B\}$ is a finite set of *adjunction features*, $F_{+} = \{+\mathtt{f} \mid \mathtt{f} \in F\}$ is a finite set of *licensor features*, $F_{-} = \{-\mathtt{f} \mid \mathtt{f} \in F\}$ is a finite set of *licensee features*, and $B_{\sim} = \{\sim\mathtt{b} \mid \mathtt{b} \in B\}$ is a finite set of *scrambling licensee features*, respectively, $c \in B$ a start category, and $\mathcal{P}$ a finite set of unary and binary operations shown below.

$$\mathcal{P} = \{\textbf{move}, \textbf{scramble}, \textbf{assume}, \textbf{merge}, \textbf{adjoin}, \textbf{discharge}\}$$

4 Analysis

Kobele's unified approach gives us multiple ways to derive an English sentence such as *Who criticized Diego?* For instance, *who* in the sentence can be inserted at different timings, providing three possible derivations. In the case of Japanese, we argue that the possibility of co-indexation of the wh-phrase and *pro* in the island is correlated with the point at which it enters the derivation. Our generalization is given in (16).

[1] This restriction may seem to spoil an analysis in the original work on slash feature percolation (Kobele, 2012). However, if the overt QR proposed by Hornstein (1995) is adopted, our proposal does not affect his analysis.

file: tozi$_{<d,V}$
v: $\epsilon_{<V,<d,v}$
Taro-TOP: Taroo-wa$_{d,-top}$
T: ta$_{<v,t}$
C$_Q$: no$_{<t,+top,+q,c}$
what-ACC: nani-o$_{d,\hat{}t,-q}$
what-ACC: nani-o$_{d,-q}$

Figure 1: Example of Japanese lexicon

(16) A *pro* may have the same index as that of a wh-phrase when the wh-phrase c-commands it when it **first** entered the derivation.

We now show how our grammar can produce our sentences of interest. The contrast to be shown is the one seen between (17) and (18). In (17), there is no movement of the wh-phrase, and the sentence is ungrammatical under the co-indexed reading. In (18), the wh-phrase is moved and therefore we can get the parasitic gap interpretation.

(17) * Taroo-wa [pg_i yomazu-ni]
Taro-TOP read.NEG-with
nani$_i$-o tozita no?
what-ACC filed Q
'What$_i$ did Taro file without reading it$_i$.'

(18) Taroo-wa nani$_i$-o [pg_i
Taro-TOP what-ACC
yomazu-ni] t_i tozita no?
read.NEG-with file Q
'What$_i$ did Taro file without reading it$_i$?'

We introduce our lexicon to derive toy-set examples of Japanese sentences given in Figure 1. Here, we ignore case features as they are irrelevant to our discussion.

We assume a subordinate small clause with a parasitic gap in (19) to simplify the discussion.

(19) ***pro* read.without**: [*pro* yomazuni]$_{»v}$

The derivation steps and a derivation tree in each step of (17), the sentence without scrambling, are shown in Figure 2 and Figure 3, respectively.

First, let us show how the derivation proceeds using derivation trees in Figure 3. (17), which does not involve the overt movement of the wh-phrase, cannot have a co-indexed reading. There are two possible derivations to get this sentence.

The first possibility is the case where the wh-phrase is merged with the verb first, as shown in Figure 3. This sentence can never obtain the co-indexed reading due to (16), i.e., the wh-phrase enters earlier than a *pro* and cannot c-command it. The derivation itself can converge, but the sentence does not have the "parasitic gap" interpretation.

The second possibility is that the object wh-phrase is assumed. In this case, our definition of **discharge** requires the assumption to be discharged immediately because the wh-phrase used in this derivation only has the -q- feature. As a result, we have virtually the same structure as in Figure 3. Consequently, the wh-phrase can never c-command the *pro* in the subordinate small clause; hence, the parasitic gap reading is unavailable.

Now let us see the derivation in detail. In Figure 2, because the wh-phrase *nani-o* is immediately merged with the verb in step 1, it cannot c-command the *pro* until it undergoes covert movement in step 8. Although we can apply **assume** to the verb in step 1, the assumption should only have the -q-feature to get the desired word order. In this case, our proposal obligates **discharge** to be applied immediately after step 1. That is, the wh-phrase *nani-o* cannot wait for the subordinate clause to be adjoined. In addition, the final result with the covert movement of the wh-phrase creates a WCO environment.

Now, let us consider the grammatical case (18), where the wh-phrase is moved from the base-generated position. The derivation steps and a derivation tree in each step of (18) are shown in Figure 4. and Figure 5 respectively. To allow *pro* to be co-indexed with the wh-phrase, the object DP must be assumed first. After adjoining the *without*-clause and merging T, the assumption is scrambled. After this clause-internal scrambling, the wh-phrase is

1. **merge**(nani-o$_{\texttt{d},\texttt{-q}}$, tozi$_{\texttt{<d},\texttt{V}}$) = [nani-o$_{\texttt{-q}}$ tozi]$_{\texttt{V}}$
2. **merge**(1, $\epsilon_{\texttt{<V},\texttt{<d},\texttt{v}}$) = [[nani-o$_{\texttt{-q}}$ tozi] ϵ]$_{\texttt{<d},\texttt{v}}$
3. **merge**(Taroo-wa$_{\texttt{d},\texttt{-top}}$, 2) = [Taroo-wa$_{\texttt{-top}}$ [[nani-o$_{\texttt{-q}}$ tozi] ϵ]]$_{\texttt{v}}$
4. **adjoin**(3, *pro* yomazuni$_{\texttt{»v}}$) = [[*pro* yomazuni] [Taroo-wa$_{\texttt{-top}}$ [[nani-o$_{\texttt{-q}}$ tozi] ϵ]]]$_{\texttt{v}}$
5. **merge**(4, ta$_{\texttt{<v},\texttt{t}}$) = [[[*pro* yomazuni] [Taroo-wa$_{\texttt{-top}}$ [[nani-o$_{\texttt{-q}}$ tozi] ϵ]]] ta]$_{\texttt{t}}$
6. **merge**(5, no$_{\texttt{<t},\texttt{+top},\texttt{+q},\texttt{c}}$)
   = [[[[*pro* yomazuni] [Taroo-wa$_{\texttt{-top}}$ [[nani-o$_{\texttt{-q}}$ tozi] ϵ]]] ta] no]$_{\texttt{+top},\texttt{+q},\texttt{c}}$
7. **move**(6) = [Taroo-wa [[[[*pro* yomazuni] [ϵ [[nani-o$_{\texttt{-q}}$ tozi] ϵ]]] ta] no]$_{\texttt{+q},\texttt{c}}$
8. **move**(7) = [ϵ [Taroo-wa [[[[*pro* yomazuni] [ϵ [[nani-o tozi] ϵ]]] ta] no]]]$_{\texttt{c}}$

Figure 2: Derivation of *Taroo-wa yomazuni nani-o tozitano?*

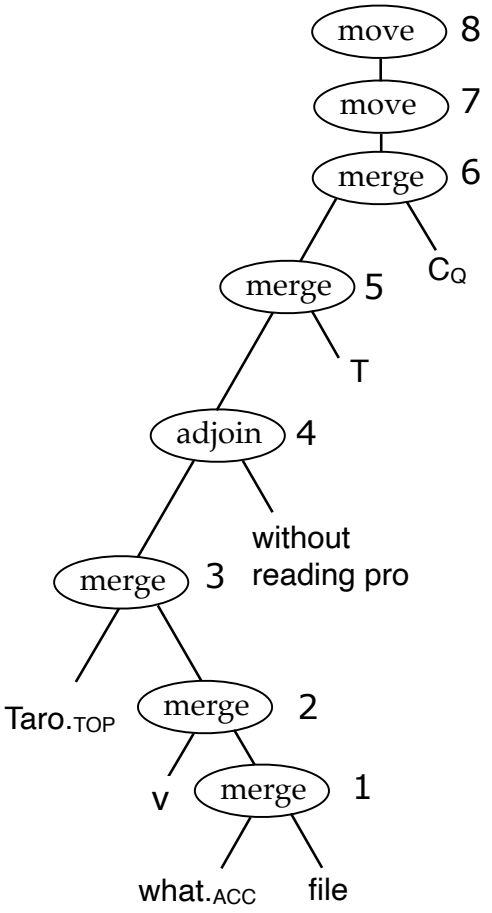


Figure 3: Derivation trees for the sentences without scrambling

discharged. In the final part of the derivation, the q-feature is checked by covert movement. On the surface, we have a weak crossover configuration, but the weak crossover violation is remedied thanks to clause-internal scrambling. In summary, clause-internal scrambling, as A-movement (Saito, 1992), can license null pronouns appearing as parasitic gaps (18) as well as overt pronouns (20).

(20) Taroo-wa nani$_i$-o [soitu$_i$-no
Taro-TOP what-ACC its$_i$
kabaa-goto] t_i tozita no?
cover-with filed Q
'What$_i$ did Taro file t_i with its$_i$ cover?'

In other words, though wh-configuration bleeds the licensing of the co-indexed readings of null pronouns, scrambling can counterbleed licensing of parasitic gaps.

Figure 4 shows that because the assumption is to be scrambled, it has the feature ˜t in step 1 and can wait to be discharged later in the derivation in step 7. Consequently, the wh-phrase *nani-o* can c-command the *pro* when it first enters the derivation in the same step.

5 Discussion

Here, we discuss our proposal and compare it with other previous studies on MGs and parasitic gaps. First, the grammar used in this paper is compared with those in the previous studies in terms of the plausibility of the extension. Next, we examine how the proposed analysis is different from (i) the previous studies on parasitic gps using MGs and (ii) Hirayama (2018).

1. **assume**(tozi$_{<\mathtt{d},\mathtt{V}}$) = [[$\mathtt{d}$, ˜$\mathtt{t}$, $-\mathtt{q}$]$_{\tilde{}\mathtt{t},-\mathtt{q}}$ tozi]$_{\mathtt{V}}$
2. **merge**(1, $\epsilon_{<\mathtt{V},<\mathtt{d},\mathtt{v}}$) = [[[$\mathtt{d}$, ˜$\mathtt{t}$, $-\mathtt{q}$]$_{\tilde{}\mathtt{t},-\mathtt{q}}$ tozi] ϵ]$_{<\mathtt{d},\mathtt{v}}$
3. **merge**(Taroo-wa$_{\mathtt{d},-\mathtt{top}}$, 2) = [Taroo-wa$_{-\mathtt{top}}$ [[[$\mathtt{d}$, ˜$\mathtt{t}$, $-\mathtt{q}$]$_{\tilde{}\mathtt{t},-\mathtt{q}}$ tozi] ϵ]]$_{\mathtt{v}}$
4. **adjoin**(3, [*pro* yomazuni]$_{\gg\mathtt{v}}$) = [[*pro* yomazuni] [Taroo-wa$_{-\mathtt{top}}$ [[[$\mathtt{d}$, ˜$\mathtt{t}$, $-\mathtt{q}$]$_{\tilde{}\mathtt{t},-\mathtt{q}}$ tozi] ϵ]]]$_{\mathtt{v}}$
5. **merge**(4, ta$_{<\mathtt{v},\mathtt{t}}$) = [[[[*pro* yomazuni] [Taroo-wa$_{-\mathtt{top}}$ [[[$\mathtt{d}$, ˜$\mathtt{t}$, $-\mathtt{q}$]$_{\tilde{}\mathtt{t},-\mathtt{q}}$ tozi] ϵ]]] ta]$_{\mathtt{t}}$
6. **scramble**(5) = [[$\mathtt{d}$, ˜$\mathtt{t}$, $-\mathtt{q}$]$_{-\mathtt{q}}$ [[[*pro* yomazuni] [Taroo-wa$_{-\mathtt{top}}$ [[ϵ tozi] ϵ]]] ta]]$_{\mathtt{t}}$
7. **discharge**(6, nani-o$_{\mathtt{d},\tilde{}\mathtt{t},-\mathtt{q}}$) = [nani-o$_{-\mathtt{q}}$ [[[*pro* yomazuni] [Taroo-wa$_{-\mathtt{top}}$ [[ϵ tozi] ϵ]]] ta]]$_{\mathtt{t}}$
8. **merge**(7, no$_{<\mathtt{t},+\mathtt{top},+\mathtt{q},\mathtt{c}}$)
   = [[nani-o$_{-\mathtt{q}}$ [[[*pro* yomazuni] [Taroo-wa$_{-\mathtt{top}}$ [[ϵ tozi] ϵ]]] ta]] no]$_{+\mathtt{top},+\mathtt{q},\mathtt{c}}$
9. **move**(8) = [Taroo-wa [[nani-o$_{-\mathtt{q}}$ [[[*pro* yomazuni] [ϵ [[ϵ tozi] ϵ]]] ta]] no]]$_{+\mathtt{q},\mathtt{c}}$
10. **move**(9) = [ϵ [Taroo-wa [[nani-o [[[*pro* yomazuni] [ϵ [[ϵ tozi] ϵ]]] ta]] no]]]$_{\mathtt{c}}$

Figure 4: Derivation of *Taroo-wa nani-o yomazuni tozitano?*

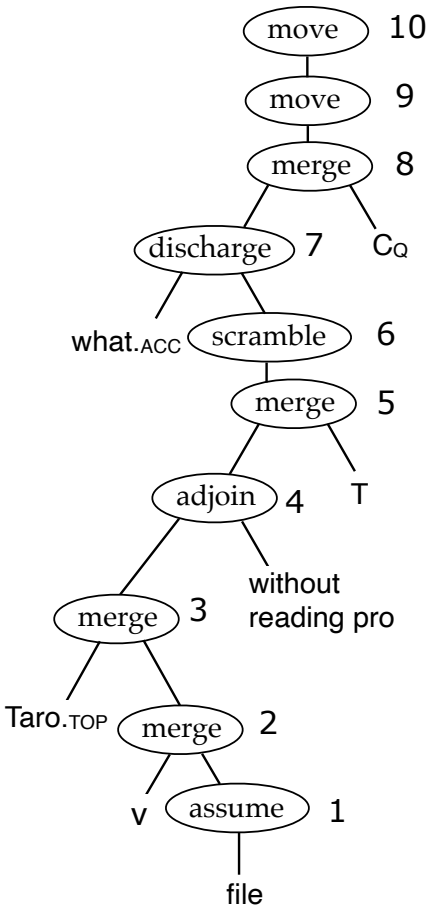


Figure 5: Derivation trees for the sentences with scrambling

5.1 Comparison with the grammars proposed in the previous work

We incorporate several operations for syntax such as **scrabling** or **adjunction**, in addition to **merge** and **move**. The increased number of operations makes the MG's generative capacity obscure. However, it is worth mentioning that the operations are not motivated only specifically for Japanese. Frey and Gärtner (2002) introduce **scrambling** and **adjunction** to treat some phenomena in Indo-European languages. We adopt these operations for the analysis of the phenomenon in non-Indo-European languages. In addition, these operations do not increase derivational ambiguity for a single sentence, as they are driven by features different from **merge** and **move**.

In contrast, **assume** and **discharge** proposed by Kobele (2012) seem unwelcome in some sense that these operations may inconstantly increase the number of ambiguous derivations for a single sentence. However, we reformulated **discharge** (15) in a more restricted way than the original definition; Our definition states that **discharge** must be applied only to the pair of the syntactic context whose specifier position is occupied by the dummy object with an unsaturated feature, and the corresponding object. This leads to a reduction of some ambiguous courses of derivations.

5.2 Comparison with previous work on parasitic gaps with MG

Stabler (2006) and Kobele (2008) proposed MG-based analyses (or equivalent formalisms based) for parasitic gaps in English. Both adopted a derivational model similar to sideward movement. In contrast, our approach is more representational.

5.3 Comparison with Hirayama (2018)

In Section 2.1, we mentioned the approach of Hirayama (2018) is semantic. Her analysis assumes no LF movement of the wh-phrase, and movement is necessary so that a single lambda can bind both *pro* and the trace of the wh-phrase via Predicate Abstraction, as schematically illustrated in (21). After the lambda binds both *pro* and the trace, the wh-phrase can manipulate both values simultaneously. Without a trace, namely, when the wh-phrase stays in situ, it cannot affect the value of *pro* in the semantic computation process.

(21) who ... λ_3 ... [... pro_3] ... t_3

Hirayama's analysis is problematic in explaining the case with the subject wh-phrase. As mentioned earlier, the anti-c-command condition does not hold in Japanese. In other words, the real trace can c-command a parasitic gap in Japanese, as seen in (22):

(22) Dono gakusee-ga Hanako-ni
which student-NOM Hanako-by
[Taroo-ga pg_i sagasu mae-ni]
Taroo-NOM look for before
mitukatta no?
found Q
'Which student$_i$ t_i got found by Hanako before Taro looked for pg_i?'

For Hirayama's semantic analysis to work, there should be a trace of the wh-subject so that we can have the configuration in (21). As she mentions in footnote 9, it is possible to assume a vacuous clause-internal movement of the subject. However, as this is not a weak crossover configuration, nothing motivates the vacuous movement.

By contrast, our analysis only refers to the steps in the derivation to account for the possibility of co-indexation. In the case of (22), the subordinate clause has already entered the derivation, so the subject wh-phrase c-commands it when it enters the derivation. The subject wh-phrase covertly checks the q-feature, but there is no need to assume further scrambling because it is not a weak crossover configuration.

6 Conclusion

We proposed a DMG-based analysis of parasitic gaps in Japanese, using a slash-feature percolation. Though we had observed several exotic properties of parasitic gaps in Japanese, a typical example of non-Indo-European languages, we have shown an extension of the 'minimalist' assumptions to deal with them precisely. The interaction between **discharge** and **scramble** explains that A-movement can counterbleed the WCO effect.

The proposed DMG $G_{\mathcal{P}}$ contains six operations, which makes the generative capacity of this grammar unclear. However, because some properties of the parasitic gaps in Japanese can be explained in the interaction of these operations, these operations appear to be necessary to account for some properties of natural language. More work on a variety of languages by MG would be needed to seek "minimalist" grammar.

Acknowledgment

We are grateful to Johannes Schneider for his helpful comments and discussion. This study is supported by JSPS KAKENHI Grant Number 21K12985 (PI: Hitomi Hirayama).

References

Jun Abe. 2011. Real parasitic gaps in Japanese. *Journal of East Asian Linguistics*, 20(3):195–218.

Noam Chomsky. 1995. *The Minimalist Program*. The MIT press, Cambridge, MA.

Elisabet Engdahl. 1983. Parasitic gaps. *Linguistics and Philosophy*, 6(1):5–34.

Werner Frey and Hans-Martin Gärtner. 2002. On the Treatment of Scrambling and Adjunction in Minimalist Grammars. In *Proceedings of formal grammar*, pages 41–52, Trento, Italy. FGTrento.

Hitomi Hirayama. 2018. Revisiting a null pronominal account for parasitic gaps in Japanese. *Glossa: a journal of general linguistics*, 3(1).

Norbert Hornstein. 1995. *Logical form: from GB to minimalism*. Generative syntax. Blackwell, Oxford, UK ; Cambridge, USA.

Richard S. Kayne. 1994. *The Antisymmetry of Syntax*. Number 25 in Linguistic inquiry monographs. MIT Press.

Gregory M. Kobele. 2008. Across-the-Board Extraction in Minimalist Grammars. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammar and Related Frameworks*, pages 113–120, Tübingen, Germany.

Gregory M. Kobele. 2010. A Formal Foundation for A and A-bar Movement. In Christian Ebert, Gerhard Jäger, and Jens Michaelis, editors, *The Mathematics of Language*, volume 6149, pages 145–159. Springer Berlin Heidelberg, Berlin, Heidelberg.

Gregory M. Kobele. 2012. Deriving Reconstruction Asymmetries. In Artemis Alexiadou, Tibor Kiss, and Gereon Müller, editors, *Local Modelling of Non-Local Dependencies in Syntax*, pages 477–500. De Gruyter.

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

Junko Shimoyama. 2006. Indeterminate Phrase Quantification in Japanese. *Natural Language Semantics*, 14(2):139–173.

Edward P. Stabler. 1997a. Computing Quantifier Scope. In Gennaro Chierchia, Pauline Jacobson, Francis J. Pelletier, and Anna Szabolcsi, editors, *Ways of Scope Taking*, volume 65, pages 155–182. Springer Netherlands, Dordrecht. Series Title: Studies in Linguistics and Philosophy.

Edward P. Stabler. 1997b. Derivational minimalism. In *Logical aspects of computational linguistics*, pages 68–95, London, UK. Springer-Verlag.

Edward P Stabler. 2006. Sidewards without copying. In Shuly Wintner, editor, *Formal Grammar 2006*, pages 157–170. CSLI, Malaga, Spain.

Edward P. Stabler. 2011. Computational Perspectives on Minimalism. In Cedric Boeckx, editor, *The Oxford Handbook of Linguistic Minimalism*. Oxford University Press.

Daiko Takahashi. 2006. Apparent Parasitic Gaps and Null Arguments in Japanese*. *Journal of East Asian Linguistics*, 15(1):1–35.

Noriko Yoshimura. 1992. *Scrambling and anaphora in Japanese*. Ph.D. thesis, University of Southern California, Los Angeles, CA.

Parsing "Early English Books Online" for Linguistic Search

Seth Kulick and **Neville Ryant**
Linguistic Data Consortium
University of Pennsylvania
{skulick,nryant}@ldc.upenn.edu

Beatrice Santorini
Department of Linguistics
University of Pennsylvania
beatrice@sas.upenn.edu

Abstract

This work addresses the question of how to evaluate a state-of-the-art parser on Early English Books Online (EEBO), a 1.5-billion-word collection of unannotated text, for utility in linguistic research. Earlier work has trained and evaluated a parser on the 1.7-million-word Penn-Helsinki Parsed Corpus of Early Modern English (PPCEME) and defined a query-based evaluation to score the retrieval of 6 specific sentence types of interest. However, significant differences between EEBO and the manually-annotated PPCEME make it inappropriate to assume that these results will generalize to EEBO. Fortunately, an overlap of source material in PPCEME and EEBO allows us to establish a token alignment between them and to score the POS-tagging on EEBO. We use this alignment together with a more principled version of the query-based evaluation to score the recovery of sentence types on this subset of EEBO, thus allowing us to estimate the increase in error rate on EEBO compared to PPCEME. The increase is largely due to differences in sentence segmentation between the two corpora, pointing the way to further improvements.

1 Introduction

The Penn-Helsinki Parsed Corpus of Early Modern English (PPCEME) (Kroch et al., 2004) consists of over 1.7 million tokens of text from 1500 to 1712, manually annotated for phrase structure. It belongs to a family of treebanks of historical English (Taylor et al., 2003, 2006; Kroch, 2020) and other languages (Wallenberg et al., 2011; Galves et al., 2017; Martineau et al., 2021; Kroch and Santorini, 2021) with a shared annotation philosophy and similar guidelines across languages, which form the basis for reproducible studies of syntactic change (Kroch et al., 2000; Ecay, 2015; Wallenberg, 2016; Galves, 2020; Wallenberg et al., 2021).

While all of these corpora are large for manually annotated corpora, even relatively common phenomena still occur too rarely to support reliable statistical models of how they change over time. We therefore wish to parse and search the much larger corpora that are becoming publicly available, such as the Early English Books Online (EEBO) corpus (Text Creation Partnership, 2019) with its 1.5 billion words of text from 1475 to 1700. However, EEBO's potential as a resource for linguistic research remains unrealized because it is not linguistically annotated and its size renders manual annotation infeasible. Our goal is therefore to parse EEBO automatically.

Kulick et al. (2022a) took first steps in this direction by training and evaluating a constituency parser using the gold trees from PPCEME. This parser achieved a cross-validated evalb score (Sekine and Collins, 2008) of 90.53%, suggesting the feasibility of the larger project of parsing EEBO. In a follow-up paper, Kulick et al. (2022b) directly evaluated the utility of the recovered parse trees for the retrieval of sentence types necessary to study a particular linguistic change in the history of English. Utilizing a novel alternative to evalb, termed "query-based evaluation", the parser was evaluated by specifically scoring the retrieval of these sentence types. The resulting precision scores were promising, warranting further work.

However, Kulick et al. (2022a,b) obtained their results for PPCEME, not EEBO. While both corpora consist of Early Modern English texts, they harbor significant differences, making it inappropriate to assume that results obtained for PPCEME generalize to EEBO.

In this work, we therefore extend the parser evaluation to EEBO itself. An apparently intractable difficulty is the absence of gold parse trees for EEBO. Fortunately, there is some overlap between PPCEME and EEBO; specifically, about 42% of PPCEME consists of source texts also present in EEBO, though possibly based on variant editions that differ in spelling and punctuation. Using an improved language model, we train a parser

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 222-242.
Amherst, Massachusetts, June 15-17, 2023

Missing words or punctuation:

```
haue alway resysted hym , and
haue       resisted       and
```

Tokenization differences:

```
In whom , nat withstandyng   ,
In whom   not with standynge ,
```

Bullet (illegible) character in EEBO:

```
I will not let , openlie to
I will not l•• , openlie to
```

Figure 1: Examples of mismatches in PPCEME (top) and EEBO (bottom) source texts.

on the non-overlap section of PPCEME and then parse both the PPCEME and EEBO versions of the overlap. We create a token alignment between the two overlap versions, which allows us to evaluate the parsed EEBO overlap for part-of-speech (POS) accuracy. We also improve the mechanics of the query-based evaluation from Kulick et al. (2022b) and use that, together with the alignment, to evaluate the parser's performance on the EEBO overlap text.

The rest of the paper is structured as follows. Section 2 discusses some important features of the overlap and the alignment between the two versions. Section 3 presents the parser model, along with results on PPCEME based on evalb, which we include to show improvements due to the new language model. Section 4 discusses the parsing of the EEBO overlap and the POS evaluation. Section 5 describes the queries and the new alignment-mediated scoring method, and Section 6 presents the results. Section 7 summarizes with lessons learned and suggestions for future work.

2 PPCEME-EEBO Overlap

2.1 Overview

PPCEME consists of material from 232 source texts, 42 of which have EEBO counterparts (see Appendix A for details). It might be thought that PPCEME should form a proper subset of EEBO, but this is not the case as while EEBO consists of all English-language material printed before 1700, many texts in PPCEME - notably private letters and editions of minor plays - did not appear in print until after 1700.

Figure 1 illustrates the main differences between the PPCEME and EEBO versions of the overlap.

source	# sents	# tokens	tokens/sent
PPCEME	39,400	805,475	20.44
EEBO	28,378	813,947	28.68

Table 1: Sentence counts and token counts for the PPCEME and EEBO versions of the overlap material.

The first example shows how one version may have tokens that are entirely missing from the other ("alway", "hym", ","). The second shows a typical case of a whitespace tokenization difference - "withstandyng" vs. "with standynge". Both examples also show differences in spelling and punctuation. The third example is a very specific type of spelling difference. Illegible characters in the source material are represented in EEBO by a bullet character. Here, "l••" has two illegible characters, corresponding to "let" in PPCEME.[1]

A further significant difference concerns sentence segmentation. Sentence segmentation in PPCEME was performed manually in accordance with annotation guidelines based on linguistic considerations. This is not the case for EEBO. The lack of standardized punctuation conventions for Early Modern English makes it non-trivial to segment sentences according to modern punctuation conventions, let alone according to PPCEME's guidelines.

Figure 2 gives two examples, each of which illustrates a string of text divided into separate sentences (and therefore trees) in PPCEME. The corresponding nearly identical text in EEBO is not so divided. As is evident, PPCEME sometimes splits on commas (e.g., the comma after *rest*) and colons (e.g., the colon after *Gold*). In contrast, after word tokenization, we split sentences in EEBO automatically on question mark, exclamation mark, and period.[2] (The reason that EEBO has no sentence break after *quills* in the first example is that it has a comma for PPCEME's period.) We refrain from splitting on commas because doing so would massively overgenerate sentence fragments.

As a result, sentences in EEBO tend to be longer than in PPCEME. Table 1 shows the number of tokens, sentences, and mean sentence length for the overlap material. The number of tokens is roughly the same, but EEBO has fewer sentences and so higher mean sentence length. Appendix A breaks

[1] PPCEME contains no bullet characters because any illegible characters were manually resolved in the process of either data entry or annotation.

[2] We do not split on periods in common abbreviations, Roman numerals of the era, and the like.

```
|| His Dame comming home and hearing that her man was gone to bed , tooke that night
but small rest , || and early in the morning hearing him vp at his worke merrily
singing , shee by and by arose , || and in seemely sort attyring her selfe , she
came into the worke-shop , || and sat her downe to make quills .  || Quoth Iohn ,
Good morow Dame , || how do you to day ?  ||

|| No , Nan Winchcombe , I will call her name , plaine Nan :  || what , I was a
woman , when she was sir-reuerence a paltry girle , though now shée goes in her Hood
and Chaine of Gold :  || what care I for her ?  ||
```

Figure 2: Sentence segmentation in PPCEME (indicated by vertical bars). Each of the two corresponding examples in EEBO is treated as a single long sentence.

source	# aligned	# unaligned	%
PPCEME	796,704	8,771	98.9
EEBO		17,243	97.9

Table 2: Number of aligned and unaligned tokens, and percentage of aligned, in PPCEME and EEBO.

down Table 1 by source text, revealing significant differences for some files, and also compares the sentence lengths of the PPCEME and EEBO version of the overlap to that of PPCEME and EEBO as a whole.

2.2 Alignment

The rest of the work relies on having a token-to-token (words and punctuation) alignment between the PPCEME and EEBO versions of the overlap. Both versions required some preparatory work before running our token-alignment algorithm.

For EEBO, we followed the same procedure as detailed in Kulick et al. (2022a,b) in connection with using EEBO for language model training, with sentence segmentation as just described.

In PPCEME, the 42 source texts are generally represented by non-exhaustive samples. Moreover, because of how the corpus was constructed over time, these samples do not always appear in the order in which they appear in the edition (for instance, parts of a play's fourth act might be interleaved with the first act). We therefore prepared a normalized version of the material with the sentences in order, which was then processed further as described for PPCEME in Kulick et al. (2022a,b).

We then aligned each of the 42 texts at the token level with our implementation of the Smith-Waterman algorithm (Smith et al., 1981), using a similarity measure based on Levenshtein distance (Levenshtein et al., 1966). To help anchor the alignment, we lowered the substitution costs for the bullet character (to 0.1) and the relatively common *u*/*v* alternation (to 0.2). We also forced the similarity to equal 1 for consistent cases of alternation between PPCEME and EEBO (e.g. *&c*/*etc.*, *&*/*and*, and *the*/*ye*).

Table 2 summarizes the completeness of the alignment, showing number of aligned and unaligned tokens in PPCEME and EEBO. For example, *alway*, *hym* and the comma in the first example in Figure 1 are unaligned PPCEME tokens. In the second example, *withstandyng* and *standynge* are aligned, so *with* is an unaligned token in EEBO. For additional alignment details, including per-text statistics, consult Appendix B.

3 Model and Evaluation

3.1 Parser Architecture

We use the same parser architecture as Kulick et al. (2022a,b), but with an improved language model. The parser model is based on Kitaev et al. (2019), which represents a constituency tree T as a set of labeled spans (i, j, l), where i and j are a span's beginning and ending positions and l is its label. Each tree is assigned a score $s(T)$, which is decomposed as a sum of per-span scores:

$$s(T) = \sum_{(i,j,l) \in T} s(i, j, l) \qquad (1)$$

The per-span scores $s(i, j, l)$ themselves are assigned using a neural network that takes a sequence of embeddings as input, processes these embeddings using a transformer-based encoder (Vaswani et al., 2017), and produces a span score from an MLP classifier (Stern et al., 2017). The highest-scoring valid tree is then found using a variant of the CKY algorithm. POS tags are recovered using a separate classifier operating on top of the encoder output, which is jointly optimized with the span

section	parser	POS
dev	92.08 (1.6)	98.23 (0.7)
test	91.77 (0.6)	98.37 (0.3)

Table 3: Cross-validation parser and POS results. Each result is the mean for the section (dev or test) over the 8 splits (standard deviation in parentheses). All scores are expressed as percentages.

classifier. For more details, see Kitaev and Klein (2018).

Our implementation is based on version 0.2.0 of the Berkeley Neural Parser[3], with some modifications for using the PPCEME and EEBO data as input.[4] While the earlier work used ELMo embeddings pretrained from scratch on EEBO, here we use RoBERTa embeddings (Liu et al., 2019) with continued pre-training for two epochs on EEBO starting from *roberta-base*.[5] For more details on training and hyperparameters, see Appendix C.

3.2 Cross-Validation Results on PPCEME

We use the same 8-fold split of PPCEME as in Kulick et al. (2022a,b), training each of the 8 models for 50 epochs and using the evalb score on the dev section as our criterion for saving the best model. Table 3 gives our parsing and POS results, combined over the 8 cross-validation splits, as scored by evalb (matching brackets for the parsing score and POS accuracy for the tagging score).[6]

The parser scores are all 1.2% higher (absolute) than the ELMo-based results reported in Kulick et al. (2022b), with the POS results also showing a slight increase (an average of 0.08). Kulick et al. (2022b) point out some differences in annotation style from the Penn Treebank (PTB) (e.g., lack of base NPs) that lead to lower parser scores here than if run on PTB. For details of the cross-validation splits, see Appendix D.

4 Parsing and POS Accuracy for Overlap

At this point, we have the token alignment between the PPCEME and EEBO overlap versions, and we have trained and evaluated on all of PPCEME with cross-validation, showing improved results over earlier work. Our next step is to train the parser in order to evaluate on the overlap versions.

We reserve the overlap for testing and partition the remaining non-overlap PPCEME material into training and dev sections, as set out in Table 4.

section	# files	# tokens	% of split
train	184	1,041,352	54.58
dev	6	60,960	3.20
overlap	42	805,475	42.22
total	232	1,907,787	100.00

Table 4: Split of PPCEME for evaluating on overlap.

	# tokens	parser	POS
PPCEME overlap			
all tokens	805,475	91.64	98.26
EEBO overlap			
aligned tokens	796,704	-	95.17
non-punc only	702,464	-	97.25
only w/ bullet	2,057	-	80.12

Table 5: Parser (evalb f1) and POS (accuracy) scores for PPCEME and EEBO versions of overlap.

4.1 Scoring the PPCEME overlap

Having trained the parser, we now evaluate it on the PPCEME version of the overlap. Since we have gold trees for this material, we can do so with evalb. The top part of Table 5 shows aggregate evalb and POS results. Appendix E gives a breakdown by text, including recall and precision.

The parser score of 91.64% is lower than the cross-validated results in Table 3. This is hardly surprising, since the parser is only being given 55% as much training data.

4.2 Scoring the EEBO overlap

For the EEBO version of the overlap, we have no corresponding gold trees, and so cannot evaluate with evalb.[7] However, we can - for the first time - evaluate POS accuracy on EEBO by taking advantage of the token alignment discussed in Section 2.2. 97.9% (796,704) of the tokens in EEBO are aligned to a corresponding token in PPCEME. For these tokens, we can take the gold tag in EEBO to be that of its PPCEME partner. EEBO tokens

[3] https://github.com/nikitakit/self-attentive-parser

[4] These modifications and other relevant software will be made available at https://github.com/skulick/emeparse.

[5] https://huggingface.co/roberta-base

[6] For reasons discussed in Kulick et al. (2022b), we use the modified evalb supplied with the Berkeley parser (Seddah et al., 2014), which does not remove words based on punctuation tags.

[7] But see the conclusion for a possible modification of evalb.

	# tokens				
tag	gold	EEBO	rec	prec	f1
N	93,720	92,513	96.82	95.57	96.19
P	91,175	91,190	98.89	98.91	98.90
,	57,992	71,966	76.91	95.44	85.18
D	62,701	62,440	99.49	99.08	99.29
PRO	52,368	52,204	99.34	99.03	99.19
CONJ	42,478	42,154	99.44	98.68	99.06
ADJ	35,769	35,480	95.93	95.16	95.54
NS	30,937	30,974	96.79	96.91	96.85
ADV	24,804	24,477	96.83	95.56	96.19
VB	22,724	22,718	97.39	97.37	97.38
.	36,415	22,274	88.70	54.26	67.33
NPR	19,277	20,210	88.36	92.64	90.45
PRO$	17,060	17,023	99.37	99.16	99.26
BEP	14,938	14,905	99.14	98.92	99.03
VAN	14,540	14,726	95.43	96.65	96.04
VBP	14,291	14,345	95.88	96.24	96.06
Q	14,044	13,998	98.75	98.43	98.59
MD	13,828	13,709	99.43	98.58	99.00
VBD	13,663	13,653	97.48	97.41	97.44
TO	10,890	10,858	99.54	99.25	99.39
total	796,704	796,704	95.17	95.17	95.17

Table 6: Breakdown by 20 most frequent tags for the 95.17% score in row "aligned tokens" of Table 5. Note that the *total* row includes all POS tags.

without an alignment partner are left out of the scoring.

The results are shown in row "aligned tokens" in Table 5. The score (95.17%) is lower than the corresponding score for the PPCEME overlap (98.26%). Table 6 breaks the score down by tag for the 20 most common tags. Appendix F presents more detailed results along two dimensions, breaking down the bottom (EEBO) part of Table 5 by overlap file and expanding Table 6 to include all tags.

4.3 Punctuation in EEBO

The third most common tag in Table 6, comma, and the 11th most common tag, period, have low scores of 85.18% and 67.33%, respectively. This is because they are often confused, which in turn follows from a combination of PPCEME's POS annotation style with the differences in sentence segmentation in PPCEME and EEBO discussed in Section 2.1. PPCEME tags all tree-final punctuation as period. For example, in the first two lines of Figure 2, the comma after *bed* is tagged as comma, while the one after *rest* is tagged as period. In contrast, the parser assigns comma to both in EEBO - a reasonable error since in the EEBO version, the second comma is not tree-final. Re-evaluating without these two tags (row "non-punc only" in Table 5) raises the accuracy to 97.25%.

Negative declarative sentences	
VERB-NOT-DECL	*They drank not the ale*
DO-NOT-DECL	*They did not drink the ale*
Negative imperatives	
VERB-NOT-IMP	*Drink not the ale*
DO-NOT-IMP	*Do not drink the ale*
Direct questions	
VERB-SBJ	*Drank they the ale?*
DO-SBJ	*Did they drink the ale?*

Table 7: Sentence types retrieved by query searches.

4.4 Tokens with Bullet Characters in EEBO

We were also curious about accuracy on tokens containing a bullet character. As the row "only w/ bullet" shows, the score for such tokens drops to 80.12%, although they are too rare to have a major effect on the overall score. The bullet character is completely missing from the training data. Augmenting that data to randomly include it would likely improve the score on these tokens.

5 Queries and Scoring

5.1 Query Types

Kulick et al. (2022b) focused on six sentence types, which are formulated as queries for tree structures in the CorpusSearch query language (Randall, 2010). We use the same queries here. Table 7 illustrates the three pairs of sentence types retrieved by the queries, along with our labels for them (see Appendix G for a full description of the sentence types). For each pair, the first sentence type is the variant dominant in 1500, and the second the variant dominant by 1700. The leading idea of the overall project is that large datasets like EEBO will eventually allow us to decide between competing conceptual models of the loss of the older variant - specifically, competition (Kroch, 1989; Zimmermann, 2017) versus drift (Karjus, 2020).

We run the queries over three sets of trees - *PPCEME-gold* (the gold trees from the release), *PPCEME-parsed* (the parsed trees of the PPCEME version of the overlap, using the parser trained with the split described in Section 4), and *EEBO-parsed* (the parsed trees of the EEBO version of the overlap, using the same parser). This allows us to ad-

dress the problem outlined in the introduction - determining the accuracy of the query-based retrieval on parsed EEBO text as compared to parsed PPCEME text - by comparing query hits on *EEBO-parsed* and *PPCEME-parsed*, respectively, to query hits on *PPCEME-gold*.

5.2 Query Scoring on PPCEME

For scoring the query retrieval on PPCEME, we can use the same approach as Kulick et al. (2022b) for scoring queries over the PPCEME cross-validation splits. Since we are comparing parsed to gold versions of the same text, the sentence segmentation and tokens are identical, and the comparison can therefore proceed on a tree basis. Each query hit is considered to have a location (tree #, index), where the tree number is the tree it occurs in, and the index is an arbitrary numbering of the number of hits within a tree (usually just 1). Since the trees are in alignment, the matches are those for which the query hits from the gold and parsed trees have the same location, and the recall/precision/f-measure scores follow.

5.3 The Need for a New Method

However, this approach does not extend to scoring *EEBO-parsed* vs. *PPCEME-gold*, since neither the sentence segmentation nor the tokens necessarily match up. Figure 3 illustrates the problem, using the last two segments from the first example in Figure 2. The left side shows two gold PPCEME trees, while the corresponding text on the right comes from one large EEBO tree, due to the different segmentation in EEBO.

The lower PPCEME tree shows a VERB-SBJ query hit covering *how do you to day ?*, and the EEBO tree fragment shows a VERB-SBJ query hit covering *Good morrow Dame , how doe you to day*. (The question mark is not part of the hit in the EEBO version, since there it is outside of the CP-QUE-MAT clause.)

While the text covered is different, both query hits correctly label the sentences they find as VERB-SBJ. But the PPCEME trees are #s 137 and 138 among the PPCEME trees, while the EEBO tree is #98 among the EEBO trees, and so comparison by tree number is not possible.

5.4 Alignment-Mediated Scoring

However, we also have the spans of the constituents from the query hits, as indicated by `<3272,3278>` for the PPCEME tree and `<3001,3009>` for the EEBO tree. These spans refer to the PPCEME or EEBO overlap section as a whole, not to the individual trees. We can use this span information, together with our word alignment, to carry out an *alignment-mediated scoring* (AMS), as follows:

(1) Given a list of m query hits from the gold trees, and n query hits from the parsed trees, we form a $m \times n$ array of scores for each pair of hits. The score for a pair of hits is computed by using the token alignment to convert the *EEBO-parsed* span to a span in *PPCEME-gold*, and then computing a simple span overlap, normalized for length. For example, in Figure 3, `<3001,3009>` in the EEBO tree maps (by the alignment) to `<3268,3277>` on PPCEME, and so the span overlap is computed between the PPCEME hit span `<3272,3278>` and the span mapped from EEBO, `<3268,3277>`. The span overlap score is 0.55, and this becomes the score for this pair of hits. Hits in PPCEME and EEBO that match exactly would have a score of 1.0, and ones with no tokens in common (again, after using the alignment to compare them) would have a score of 0.0.

(2) We treat this as a bipartite graph minimum weight matching problem, where the "weight" for a pair of trees is one minus the span overlap, computed as just described. In this way the "penalty" for the overall mapping is minimized. We filter the results to ensure that all hits have at least one token in common.

For consistency, we compare *PPCEME-parsed* to *PPCEME-gold* in the same way as *EEBO-parsed* to *PPCEME-gold* (that is, using AMS). This also demonstrates the validity of the algorithm, since the results for *PPCEME-parsed* vs. *PPCEME-gold* using AMS are identical to those using the method from Kulick et al. (2022b).[8] Further details of the algorithm are available in Appendix H.

6 AMS Results and Analysis

6.1 Results and Analysis for EEBO

Table 8 presents scores from the query-based evaluation for *PPCEME-parsed* and *EEBO-parsed*, using AMS to produce the latter results. Our goal was to estimate the effect on the query results of

[8] We thus also resolve a lingering doubt from the earlier work. The earlier scoring method allowed a hit in a parsed PPCEME tree to "match" a hit in a completely different part of the corresponding PPCEME gold tree. The current method rules out such spurious matches, since it relies on the actual spans, not just the trees in which they occur.

```
(IP-MAT
    (VBD Quoth)
    (NP-SBJ (NPR Iohn))
    (, ,)
    (QTP (INTJP (ADJ Good)
                (N morow))
         (NP-VOC (N Dame)))
    (. ,))

(CP-QUE-MAT *<3272,3278>|verb-sbj*
    (WADVP-1 (WADV how))
    (IP-SUB
        (ADVP *T*-1)
        (DOP do)
        (NP-SBJ (PRO you))
        (NP-TMP (N (N to)
                   (N day))))
    (. ?))
```

(a) Two gold PPCEME trees

```
[...]
(IP-PPL
    (VBD quoth)
    (NP-SBJ (NPR Iohn))
    (, ,)
    (CP-QUE-MAT *<3001,3009>|verb-sbj*
        (NP-VOC (ADJ Good)
                (N morrow)
                (NP-VOC (N Dame)))
        (, ,)
        (WADVP (WADV how))
        (IP-SUB
            (DOP doe)
            (NP-SBJ (PRO you))
            (NP-TMP (N (N to)
                       (N day)))))))
(. ?))
```

(b) Fragment of one parsed EEBO tree

Figure 3: Example of the query matching problem. (a) shows two gold PPCEME trees with a VERB-SBJ query hit in the lower tree, at span <3272,3278> covering *how do you to day ?*. (b) is a fragment of a larger parsed EEBO tree with a VERB-SBJ query hit at span <3001,3009> covering *Good morrow Dame , how doe you to day*.

	PPCEME-gold	*PPCEME-parsed*				*EEBO-parsed*			
query	# hits	# hits	recall	prec	f1	# hits	recall	prec	f1
Negative declarative sentences									
VERB-NOT-DECL	662	680	95.47	92.94	94.19	634	87.16	91.01	89.04
DO-NOT-DECL	329	318	95.44	98.74	97.06	304	89.97	97.37	93.52
Negative imperative sentences									
VERB-NOT-IMP	148	135	81.76	89.63	85.51	120	71.62	88.33	79.10
DO-NOT-IMP	31	26	80.65	96.15	87.72	25	77.42	96.00	85.71
Questions									
VERB-SBJ	302	266	79.47	90.23	84.51	228	68.54	90.79	78.11
DO-SBJ-ORD	306	282	90.20	97.87	93.88	253	80.72	97.63	88.37

Table 8: AMS results for *PPCEME-parsed* and *EEBO-parsed* versions of overlap, as compared to *PPCEME-gold* trees.

parsing on EEBO instead of PPCEME. This table provides the answer - the f1 scores generally decrease by about 4-6 points. (The score for DO-NOT-IMP, which is less frequent, decreases less.)

Comparing the recall and precision scores reveals that the decrease is largely due to decreases in recall. Precision stays relatively stable, while recall goes down by as much as 10 points (e.g. for VERB-SBJ, from 79.47% to 68.54%). This means that parser errors on EEBO are preventing the queries from finding the structure that is present in the gold PPCEME trees.

Examination of the parser errors suggests that longer sentence length is exacerbating a tendency of the parser (already noted in Kulick et al. (2022b)) to produce nonsensical flat structures with two subjects or two finite verbs (or both). For example, consider the second sentence in the example of sentence segmentation in Figure 2. The last segment *what care I for her ?* is a VERB-SBJ that is missed in the *EEBO-parsed* tree because the parse of the entire sentence *No , Nan Winchcombe ... for her ?* is such a nonsensical structure. Omitting details, the structure of the parse, with the two subjects and two finite verbs, is shown in Figure 4.

corpus	gold	parsed
PPCEME	4	233
EEBO	-	934

Table 9: Number of trees with two subjects, as one example of nonsensical parser error.

Parser error analysis can be complex and tedious (especially here, with the differences in sentence segmentation), but we can facilitate it by extending our use of query-based searches from finding

```
(CP-QUE-MAT
    (INTJ No)
    (NP-VOC (NPR Nan) (NPR Winchcombe))
    (, ,)
    (NP-SBJ (PRO I))
    (MD will)
    (VB call)
    (IP-SMC her name...Nan)
    (, :)
    (INTJP (WPRO what))
    (, ,)
    (NP-SBJ (PRO I))
    (BED was)
    ....
```

Figure 4: Incorrect flat parse on EEBO text.

structures of linguistic interest to finding structures that should never occur, such as clauses with two subjects. Table 9 shows the large increase of trees with such impossible structures in *EEBO-parsed*, although the number in *PPCEME-parsed* is already higher than desired.[9]

6.2 Cross-Validated Results on PPCEME

As pointed out in Kulick et al. (2022b), the parses need not be perfect for query-based search to be useful, since if an error rate can be estimated, it can be factored into the linguistic analysis. We have determined the increase in error rate when querying on EEBO rather than on PPCEME.

We are also interested in determining the error rate when querying on PPCEME. Kulick et al. (2022b) addressed this issue with the cross-validation query-based evaluation on PPCEME. However, that was using an older language model, and while here we presented improved evalb scores (Table 3), the improvements are not guaranteed to carry over to the query-based scores.

This other aspect is not our focus here, but it is part of the overall goal, and so we give some results in Appendix I, with updated cross-validated query results on the current version of the parser, along with some discussion of the impact of using less training data for the overlap split.

7 Conclusion

Exploiting the existence of an overlap between PPCEME and EEBO, we have succeeded in scoring POS tagging on EEBO and extending query-based evaluation to EEBO. Given these results, could the trees and POS-tags of a parsed EEBO be used with confidence? For those wishing to use the POS tags, we have shown that the POS tags can overall be expected to be of high accuracy (with some variation for individual tags, and excepting the punctuation issue discussed in Section 4.2). For the structure-based queries, we now have the needed estimate of the decrease in accuracy on EEBO compared to PPCEME. There are some obvious next steps to improve the parsing and query results on EEBO, and so lessen that decrease in accuracy.

The first priority is to address the problem of sentence segmentation in EEBO. We have shown in this paper why this is an important issue for parsing EEBO, and we can use the overlap to measure the effect more precisely. We will do so by using the token alignment to "fix" the sentence segmentation in the EEBO version of the overlap to be consistent with the sentence segmentation in the PPCEME version of the overlap, thus allowing us to directly measure the query accuracy on EEBO without the distorting effect of the segmentation differences and thus to estimate the latter effect.

Following this step, we see two possibilities for addressing the effects of the differences in sentence segmentation in PPCEME and EEBO. One approach modifies the training data, while leaving the EEBO segmentation as it is, by combining the PPCEME trees used for training when the text has a final comma in the text, thus approximating the EEBO segmentation. The other (preferred) approach would directly modify the EEBO segmentation by using the existing segmentation in PPCEME to train a segmenter for EEBO.

The are different directions to pursue after that point:

Improving the parser architecture. While Section 6 discussed the increase in nonsensical structures from PPCEME to EEBO, there were already too many (233) with PPCEME. It is possible that a parser model that moves away from the span-based approach of the Berkeley neural parser, using well-defined grammatical structures instead, might overcome this problem. In particular we plan to experiment with a Tree Adjoining Grammar (TAG) or related architecture (Kasai et al., 2018). This change of architecture would also allow for the recovery of the empty categories and co-indexing, which will be required as the range of linguistic inquiries expands, with the precise approach used to accomplish this depending on which architecture is chosen.

Another aspect of the parser architecture that

[9]The four occurrences in *PPCEME-gold* are annotation errors that have been corrected for the next release.

should be improved is the recovery of the function tags. As mentioned in Appendix C.2, currently we simply retain the function tags as part of atomic nonterminals for the training and parsing. While this approach works surprisingly well, it is potentially problematic for combinations of nonterminal/function tag that do not appear with frequency in the training data. One possible alternative is to integrate the function tag recovery in the current parser model analogously the POS tagging, as a separate classifier with a joint training loss.

Treebank representation. PPCEME is a phrase-structure treebank, with the associated linguistic queries reflecting that structure, and so it was natural for us to focus on phrase-structure parsing. However, it would be useful to represent PPCEME in a dependency format, so that a dependency parser could be used as well. While it might be possible to adapt one of the phrase-structure-to-dependency converters for use on PPCEME and the parsed EEBO, our preference would be for this to follow from the use of a TAG-like formalism, which is in a sense intermediate between a phrase-structure and dependency representation.

Application to other historical treebanks. As mentioned in the introduction, PPCEME is just one of a series of historical treebanks, which share annotation philosophy and guidelines. In addition to applying this work to those other treebanks, they would in turn serve as an extensive and varied testbed for evaluating the different parser models.

Query retrieval without parsing. An entirely different direction from the work described here, but with the same goal, is to use sentence embeddings derived from token embeddings, as in Arora et al. (2017), to identify the desired sentences directly, without using a parser at all. For example, it might be possible to find EEBO sentences "similar" to a given sentence, akin to an information retrieval system.

Additional types of annotation. The overlap and alignment to PPCEME can be used to evaluate the automatic annotation of other types of annotation on EEBO. For example, if PPCEME were annotated with lemmas for each token, then a lemmatizer on EEBO could be tested on the overlap section by using the existing alignment and treating the PPCEME lemmas for the aligned tokens as the gold lemmas. In addition, research in the last few years on improving word sense disambiguation (Bevilacqua et al., 2021) could be applied to the parsed EEBO, by using example sentences in the Oxford English Dictionary[10] to map each word instance to its sense usage.

Modified evalb. Finally, we stated in Section 4.2 that we cannot run evalb on parsed EEBO files in the absence of gold trees for EEBO. However, AMS opens up the possibility of doing just that. Since evalb scores matching brackets, it can be modified to match brackets using this approach instead of searching for identical spans. Such a modification could then even be adopted for material that already has matching text and sentence segmentation, allowing for a "fuzzy" evalb that can match brackets without an exact match, degrading the score if desired.

Acknowledgments

This work was supported by NSF grant BCS 13-046668, "Annotating and extracting detailed syntactic information from a 1.1-billion-word corpus." We thank the three anonymous reviewers for their insightful comments. We would also like to acknowledge here the tremendous debt, intellectual and in other ways, that this work owes to the late Prof. Anthony Kroch.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conference on Artificial Intelligence, Inc.

Aaron Ecay. 2015. *A multi-step analysis of the evolution of English do-support*. Ph.D. thesis, University of Pennsylvania.

Ryan Gabbard, Seth Kulick, and Mitchell Marcus. 2006. Fully parsing the Penn Treebank. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 184–191, New York, New York. Association for Computational Linguistics.

Charlotte Galves. 2020. Relaxed V-Second in Classical Portuguese. In Rebecca Woods and Sam Wolfe,

[10] https://www.oed.com

editors, *Rethinking Verb-Second*, pages 368–395. Oxford University Press.

Charlotte Galves, Aroldo Leal de Andrade, and Pablo Faria. 2017. Tycho Brahe Parsed Corpus of Historical Portuguese. http://www.tycho.iel.unicamp.br/~tycho/corpus/texts/psd.zip.

Andres Karjus. 2020. *Competition, selection and communicative need in language change: An investigation using corpora, computational modelling and experimentation*. Ph.D. thesis, University of Edinburgh.

Jungo Kasai, Robert Frank, Pauli Xu, William Merrill, and Owen Rambow. 2018. End-to-end graph-based TAG parsing with neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1181–1194, New Orleans, Louisiana. Association for Computational Linguistics.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Anthony Kroch. 1989. Reflexes of grammar in patterns of language change. *Language Variation and Change*, 1(3):199–244.

Anthony Kroch. 2020. Penn Parsed Corpora of Historical English. LDC2020T16 Web Download. Philadelphia: Linguistic Data Consortium. Contains Penn-Helsinki Parsed Corpus of Middle English, second edition, Penn-Helsinki Parsed Corpus of Early Modern English, and Penn Parsed Corpus of Modern British English.

Anthony Kroch and Beatrice Santorini. 2021. Penn-BFM Parsed Corpus of Historical French, version 1.0. https://github.com/beatrice57/mcvf-plus-ppchf.

Anthony Kroch, Beatrice Santorini, and Lauren Delfs. 2004. Penn-Helsinki Parsed Corpus of Early Modern English (PPCEME). CD-ROM, first edition, release 3. http://www.ling.upenn.edu/ppche/ppche-release-2016/PPCEME-RELEASE-3.

Anthony Kroch, Ann Taylor, and Donald Ringe. 2000. The Middle English verb-second constraint: A case study in language contact and language change. In Susan Herring, Lene Schoessler, and Peter van Reenen, editors, *Textual parameters in older language*, pages 353–391. Benjamins.

Seth Kulick, Neville Ryant, and Beatrice Santorini. 2022a. Penn-Helsinki Parsed Corpus of Early Modern English: First parsing results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 578–593, Seattle, Washington. Association for Computational Linguistics.

Seth Kulick, Neville Ryant, and Beatrice Santorini. 2022b. Parsing Early Modern English for linguistic search. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 143–157, online. Association for Computational Linguistics.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

France Martineau, Paul Hirschbühler, Anthony Kroch, and Yves Charles Morin. 2021. MCVF Corpus, parsed, version 2.0. https://github.com/beatrice57/mcvf-plus-ppchf.

Beth Randall. 2010. CorpusSearch 2: a tool for linguistic research. Download site: http://corpussearch.sourceforge.net/CS.html. User guide: https://www.ling.upenn.edu/~beatrice/corpus-ling/CS-users-guide/index.html.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 103–109, Dublin, Ireland. Dublin City University.

Satoshi Sekine and Michael Collins. 2008. evalb. http://nlp.cs.nyu.edu/evalb/.

Temple F Smith, Michael S Waterman, et al. 1981. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Ann Taylor, Arja Nurmi, Anthony Warner, Susan Pintzuk, and Terttu Nevalainen. 2006. Parsed Corpus of Early English Correspondence. Distributed by the Oxford Text Archive. Revised corrected version at https://github.com/beatrice57/pceec2.

Ann Taylor, Anthony Warner, Susan Pintzuk, and Frans Beths. 2003. York-Toronto-Helsinki Parsed Corpus of Old English Prose. Distributed by the Oxford Text Archive.

Text Creation Partnership. 2019. Early English Books Online. https://textcreationpartnership.org/tcp-texts/eebo-tcp-early-english-books-online/. Version 2019-04-25.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Joel C. Wallenberg. 2016. Extraposition is disappearing. *Language*, 92(4):e237–e256.

Joel C. Wallenberg, Rachael Bailes, Christine Cuskley, and Anton Karl Ingason. 2021. Smooth signals and syntactic change. *Languages*, 6(2):60.

Joel C. Wallenberg, Anton Karl Ingason, Einar Freyr Sigurdhsson, and Eiríkur Rögnvaldsson. 2011. Icelandic Parsed Historical Corpus (IcePaHC), v0.9. http://www.linguist.is/icelandic_treebank.

Richard Zimmermann. 2017. *Formal and quantitative approaches to the study of syntactic change: Three case studies from the history of English*. Ph.D. thesis, University of Geneva.

A Detailed Overlap Section Statistics

A.1 Sentence lengths by file

Table 10 extends Table 1 from Section 2 to the 42 files shared by PPCEME and EEBO. For each corresponding file (e.g., *armin-e2/A21397*), it details the total number of tokens and mean sentence length for both versions.

A.2 How representative is the overlap section?

In Figure 5 and Table 11 we provide summaries of the sentence length distributions for both PPCEME and EEBO, both overall and for the overlap section alone. From both the summary statistics and kernel density estimates (KDE), it is readily apparent that the PPCEME overlap is very representative of PPCEME overall, showing a nearly identical distribution of sentence lengths. The EEBO overlap shows less of a match to EEBO overall, with the overlap section containing a much higher proportion of extremely short sentences relative to EEBO as a whole.

This striking difference in distributions for EEBO overlap vs EEBO overall is overwhelmingly an artifact of how the overlap correspondence was constructed. As discussed in Appendix B, the EEBO version of the overlap section contains character names in plays that are counted as two-word "sentences". However, when considering all of EEBO, we only consider sentences with EEBO contexts (as indicated by the markup in the EEBO XML files) that are relevant for the query search (<P> indicating prose and <SP/L> indicating verse structure within speech.) The EEBO overall sentence lengths therefore do not include these two-word "sentences". The main point here is that the segmentation problem discussed throughout the main text is not peculiar to the EEBO overlap section. The average sentence length throughout EEBO is greater than that of PPCEME.

B Alignment Details

B.1 Alignment by Source

Table 12 expands on Table 2 by providing full alignment statistics for each text. In addition to raw counts for number of insertions/deletions/substitutions in each text, it also provides a summary statistic for alignment quality – token error rate (TER) – which is defined as:

$$TER = 100 * \frac{\text{\# insert} + \text{\# del} + \text{\# sub}}{\text{\# PPCEME tokens}} \quad (2)$$

where # insert/del/sub are the total count of insertion, deletion, and substitution errors for the text.

B.2 Alignment Algorithm Details

As mentioned in Section 2.2, the sentences of the PPCEME source texts are not always in the same order as in the corresponding EEBO files, and so we first focused on a rough correspondence between the PPCEME and EEBO versions of the overlap, followed by the word alignment. Since the sentences of the EEBO files were in the proper order, we rearranged the PPCEME sentences to match that order. At the same time, some of the meta info in PPCEME, such as character names in plays, was filtered out of the the PPCEME source by the initial preprocessing of PPCEME. As a result, the EEBO overlap has instances of character names that are not present in the PPCEME version of the overlap.

We spot-checked cases of unaligned tokens in both directions, making sure that such cases fell into the categories discussed in the text (e.g., the first two cases in Figure 1), or the character names just discussed. In addition, each pair of aligned tokens has a Levenshtein distance similarity score,

	PPCEME				EEBO			
	name	# sents	# tokens	tokens/sent	name	# sents	# tokens	tokens/sent
0	armin-e2	1,271	18,768	14.77	A21397	358	18,150	50.70
1	asch-e1	496	16,121	32.50	A21975	314	16,010	50.99
2	bacon-e2	480	20,181	42.04	A01516	260	20,209	77.73
3	behn-e3	675	19,335	28.64	A27305	302	19,481	64.51
4	blundev-e2	750	22,619	30.16	A16221	374	22,787	60.93
5	boethpr-e3	1499	32,806	21.89	A28548	1,332	33,176	24.91
6	boylecol-e3	165	7,544	45.72	A28975	78	7,545	96.73
7	brinsley-e2	656	19,710	30.05	A16865	590	19,830	33.61
8	burnetroc-e3	680	21,112	31.05	A30466	356	21,123	59.33
9	clowes-e2	905	22,500	24.86	A19029	427	21,937	51.37
10	coverte-e2	984	20,769	21.11	A19470	446	20,785	46.60
11	deloney-e2	1346	26,738	19.86	A20126	679	27,014	39.78
12	elyot-e1	514	19,157	37.27	A21287	472	19,387	41.07
13	fabyan-e1	507	19,029	37.53	A00525	518	19,023	36.72
14	fisher-e1	466	10,915	23.42	A00771	891	10,918	12.25
15	fitzh-e1	1058	18,813	17.78	A00884	550	19,068	34.67
16	fryer-e3	610	18,970	31.10	A40522	279	19,093	68.43
17	gifford-e2	1230	21,148	17.19	A01716	922	21,642	23.47
18	harman-e1	1115	19,366	17.37	A02657	372	18,026	48.46
19	hooke-e3	539	22,494	41.73	A44323	247	22,464	90.95
20	hooker-a-e2	343	9,025	26.31	A03598	233	9,043	38.81
21	hooker-b-e2	405	8,600	21.23	A03598	258	8,641	33.49
22	hoole-e3	552	21,531	39.01	A44390	364	21,527	59.14
23	jetaylormeas-e3	404	8,682	21.49	A64030	130	8,753	67.33
24	jotaylor-e2	1106	31,202	28.21	A13415	367	31,215	85.05
25	langf-e3	767	18,351	23.93	A49545	355	18,140	51.10
26	latimer-e1	966	17,603	18.22	A05143	698	17,827	25.54
27	markham-e2	253	6,138	24.26	A06913	47	6,192	131.74
28	middlet-e2	2117	19,051	9.00	A07493	3,111	21,624	6.95
29	milton-e3	638	21,307	33.40	A50902	395	21,325	53.99
30	record-e1	1092	23,422	21.45	A10541	620	23,778	38.35
31	shakesp-e2	2332	22,032	9.45	A11954	2,315	24,166	10.44
32	smith-e2	949	18,408	19.40	A12367	457	18,463	40.40
33	stevenso-e1	1512	16,936	11.20	A12969	1,962	17,404	8.87
34	stow-e2	640	17,457	27.28	A13043	353	17,627	49.93
35	turner-e1	581	16,302	28.06	A14053	464	16,320	35.17
36	turnerherb-e1	43	837	19.47	A14059	31	844	27.23
37	tyndnew-e1	2906	39,476	13.58	A68940	1,931	39,654	20.54
38	tyndold-e1	2149	33,901	15.78	A13203	1,209	34,080	28.19
39	vanbr-e3	2081	25,052	12.04	A65075	2,570	27,954	10.88
40	vicary-e1	954	19,510	20.45	A14387	424	19,269	45.45
41	walton-e3	664	12,557	18.91	A67462	317	12,433	39.22
	total	**39,400**	**805,475**	**20.44**		**28,378**	**813,947**	**28.68**

Table 10: Overlap between PPCEME and EEBO, with filename, number of tokens, number of sentences, and mean sentence length.

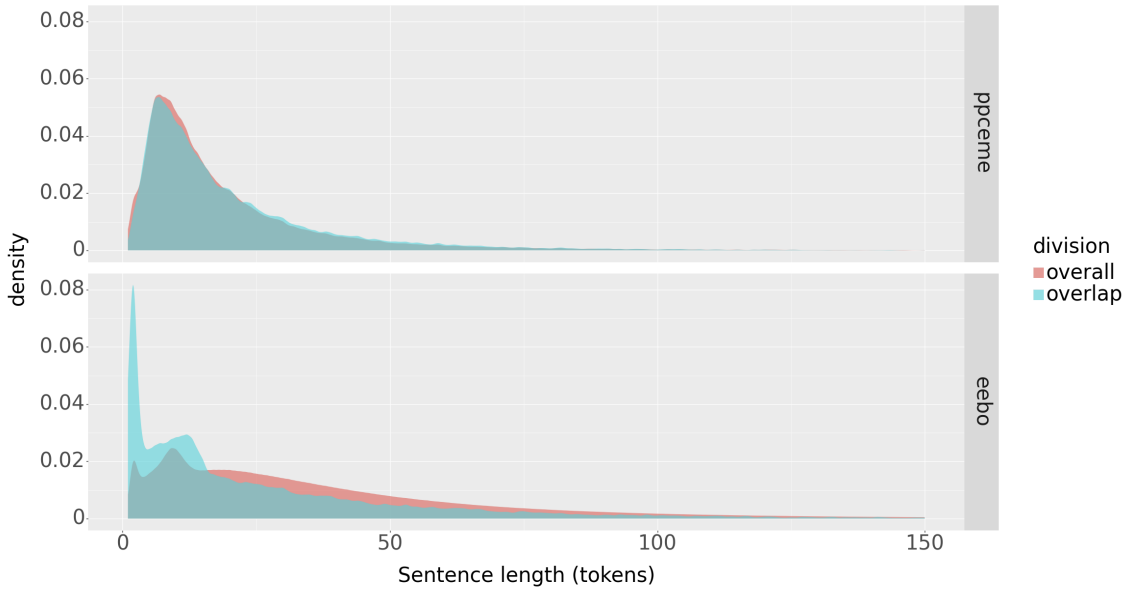


Figure 5: Kernel density estimates (KDE) for sentence length in PPCEME and EEBO. KDEs for the overall corpus and overlap section are plotted on the same figure.

			sentence length								
source	division	# sents	mean	std	min	10%	25%	50%	75%	90%	max
EEBO	overall	33,840,032	41.45	45.57	1	7	14	29	53	88	8,195
EEBO	overlap	28,378	28.68	36.89	1	2	6	16	37	69	562
PPCEME	overall	94,462	20.20	24.77	1	5	8	13	24	41	957
PPCEME	overlap	39,400	20.44	20.01	1	5	8	14	26	43	399

Table 11: Sentence length summary statistics in PPCEME and EEBO. The following statistics are presented for each corpus (both overall and for the overlap section only): mean/standard deviation, min/max, and 10/25/50/75/90-th percentiles.

	# tokens					
text	PPCEME	EEBO	TER	# insertions	# deletions	# substitutions
00-armin-e2	18,768	18,150	23.22	277	895	3,185
01-asch-e1	16,121	16,010	3.60	50	161	370
02-bacon-e2	20,181	20,209	2.16	69	41	326
03-behn-e3	19,335	19,481	19.73	262	116	3,437
04-blundev-e2	22,619	22,787	9.77	285	117	1,807
05-boethpr-e3	32,806	33,176	2.12	411	41	245
06-boylecol-e3	7,544	7,545	1.50	17	16	80
07-brinsley-e2	19,710	19,830	13.28	274	154	2,190
08-burnetroc-e3	21,112	21,123	1.17	44	33	170
09-clowes-e2	22,500	21,937	7.10	108	671	819
10-coverte-e2	20,769	20,785	3.53	40	24	670
11-deloney-e2	26,738	27,014	13.60	436	160	3,040
12-elyot-e1	19,157	19,387	24.97	630	400	3,753
13-fabyan-e1	19,029	19,023	36.82	421	427	6,159
14-fisher-e1	10,915	10,918	15.72	67	64	1,585
15-fitzh-e1	18,813	19,068	7.94	428	173	892
16-fryer-e3	18,970	19,093	2.31	160	37	242
17-gifford-e2	21,148	21,642	4.59	530	36	404
18-harman-e1	19,366	18,026	29.36	178	1518	3,990
19-hooke-e3	22,494	22,464	1.55	47	77	224
20-hooker-a-e2	9,025	9,043	1.97	45	27	106
21-hooker-b-e2	8,600	8,641	2.38	63	22	120
22-hoole-e3	21,531	21,527	2.16	91	95	279
23-jetaylormeas-e3	8,682	8,753	7.79	175	104	397
24-jotaylor-e2	31,202	31,215	3.98	141	128	972
25-langf-e3	18,351	18,140	5.70	115	326	605
26-latimer-e1	17,603	17,827	28.86	393	169	4,519
27-markham-e2	6,138	6,192	7.41	79	25	351
28-middlet-e2	19,051	21,624	17.35	2665	92	548
29-milton-e3	21,307	21,325	0.82	33	15	126
30-record-e1	23,422	23,778	11.32	554	198	1,899
31-shakesp-e2	22,032	24,166	13.41	2259	125	570
32-smith-e2	18,408	18,463	1.51	82	27	169
33-stevenso-e1	16,936	17,404	13.52	763	295	1,231
34-stow-e2	17,457	17,627	5.31	205	35	687
35-turner-e1	16,302	16,320	4.31	133	115	454
36-turnerherb-e1	837	844	10.99	7	0	85
37-tyndnew-e1	39,476	39,654	13.07	258	80	4,821
38-tyndold-e1	33,901	34,080	8.46	300	121	2,448
39-vanbr-e3	25,052	27,954	19.41	3247	345	1,270
40-vicary-e1	19,510	19,269	19.27	204	445	3,111
41-walton-e3	12,557	12,433	21.34	697	821	1,162
total	**805,475**	**813,947**	**10.62**	**17,243**	**8,771**	**59,518**

Table 12: Token alignment statistics for each text. The first two columns indicate the token counts in the PPCEME and EEBO versions of the text. The next four columns provide information about the alignment quality with *# insertions*, *# deletions*, and *# substitutions* indicating the total number of insertion, deletion, and substitution errors in the EEBO text relative to the PPCEME text given the alignment. TER is a summary statistic defined as in eqn. 2

hyperparameter	value
attention_dropout	0.2
batch_size	32
char_lstm_input_dropout	0.2
checks_per_epoch	4
clip_grad_norm	0.0
d_char_emb	64
d_ff	2048
d_kv	64
d_label_hidden	256
d_model	1,024
d_tag_hidden	256
elmo_dropout	0.5
encoder_max_len	512
force_root_constituent	'auto'
learning_rate	5e-05
learning_rate_warmup_steps	160
max_consecutive_decays	3
max_len_dev	0
max_len_train	0
morpho_emb_dropout	0.2
num_heads	8
num_layers	8
predict_tags	True
relu_dropout	0.1
residual_dropout	0.2
step_decay_factor	0.5
step_decay_patience	5
tag_loss_scale	5.0
max_epochs	50

Table 13: Hyperparameters used with the Berkeley Neural Parser.

modified by common and expected cases for character differences, as discussed in Section 2.2. We spot-checked cases where the similarity was below 0.9, which highlighted cases such as those discussed in Section 2.2 (e.g., *&* and *and*). We then treated these as special cases for the similarity metric and redid the alignment, in an iterative process.

C Model and Evaluation

Table 13 shows the hyperparameter settings used in the Berkeley Neural Parser (all default). We added a parameter `max_epochs` for the maximum number of epochs, setting it to 50 for the cross-validation training reported.

C.1 RoBERTa Pretraining

We downloaded the most recent version of English *roberta-base* from Huggingface[11] and continued pre-training for two epochs on EEBO. EEBO was preprocessed using the same steps as described in Kulick et al. (2022a,b), yielding a 1.374 billion token train set and 115K token validation set. We used the `run_mlm` script from Hugging Face with a batch size of 2 on 5 GPUs for an effective batch size of 10. Future work will explore improved performance as a function of larger models and/or additional epochs.

C.2 Function Tags

Function tags are important for us since the queries rely upon them to find the structures of linguistic interest. As in Kulick et al. (2022a,b), we adopted the approach of Gabbard et al. (2006) to function tag recovery. The function tags are retained in preprocessing, and so nonterminals like `NP-SBJ` are treated as atomic units. Since the decision whether to delete is part of preprocessing, this approach does not require modification to the parser.

C.3 Default Flat Parses

Of the 28,378 sentences in the EEBO overlap section, 5 exceeded the 512 subword limit imposed by the language model and the encoder within the parser. For such cases, we modified the parser to output a dummy flat parse, with each token assigned the tag XX.

D Cross-Validation Splits

Table 14 summarizes the composition of the train/dev/test sections across the cross-validation 8 splits; specifically, the total number of files, the total number of tokens, and the percentage of total tokens in each section. Since the partitioning process is performed at the level of PPCEME source files, and these files differ substantially in size, there is some variation in these numbers across the splits. For this reason, we report standard deviations as well as means. The final row ("total") gives numbers for a complete split (i.e., the train/dev/test sections combined); as these are constant across each split, they have a standard deviation of zero. As can be seen, overall the splits attain the target 90-5-5 breakdown; e.g., the train section on average comprises 89.65% of the total tokens with a standard deviation of 0.54%.

The total number of tokens here (1,944,480) is greater than the total number of tokens listed in Table 4 (1,907,787). This is because some sentences were removed from the PPCEME overlap files in

[11] https://huggingface.co/roberta-base

section	# files		# tokens		% of split	
train	205.88	(13.34)	1,743,211.25	(10,441.53)	89.65	(0.54)
dev	12.50	(7.15)	101,000.12	(4,081.82)	5.19	(0.21)
test	13.62	(7.91)	100,268.62	(7,832.66)	5.16	(0.40)
total	232	(0.00)	1,944,480	(0.00)	100	(0.00)

Table 14: Mean number of files and tokens for train/dev/test sections across the 8 cross-validation splits (standard deviations in parentheses). The percentage of tokens in each section is given in column "% of split".

the course of preprocessing.

E Results for PPCEME Overlap

Table 15 breaks down the scores for each of the overlap files in PPCEME. The totals for all files correspond to the number of tokens in Table 4 and the scores in the top part of Table 5.

F Results for EEBO Overlap

Here we expand in two ways on the POS tagging results on EEBO from Section 4.2. First, Table 16 breaks down the results in the bottom part of Table 5 by file. Table 17 shows the complete listing of overall results by tag, the 20 most frequent of which were shown in Table 6. The tag XX occurs in the five overly long sentences mentioned in Section C.3.

G Full Query Details

We wish to identify certain sentence types that allow us to track the rise of auxiliary *do* over the course of Early Modern English. For expository reasons, we present these sentence types in reverse chronological order.[12]

G.1 Sentence Types with Auxiliary *Do*

Modern English is unusual in requiring the auxiliary verb *do* in negative declarative sentences, negative imperatives, and all direct questions (whether positive or negative).

DO-NOT-DECL. In negative declarative sentences, the main verb appears in uninflected form. Such sentences also contain auxiliary *do* in either the present or past tense, and the negative marker *not* appears between the auxiliary and the main verb.

```
(IP-SUB (NP-SBJ (PRO they))
        (DOP do)
        (NEG not)
        (NP-MSR (Q much))
        (VB minde)
        (NP-OB1 (PRO them))
```

The IP in this sentence type (and also its historical counterpart without *do*) can be an independent matrix (MAT) clause or, as here, a subordinate (SUB) clause.

`Do-not-imp`. Negative imperatives are analogous, except for the IMP function tag on IP, and the imperative POS tag (DOI) on the auxiliary.

```
(IP-IMP (PP (P For)
            (NP (NPR$ God's)
                (N sake)))
        (DOI do)
        (NEG not)
        (VB overlay)
        (NP-OB1 (PRO me))
        (PP (P with)
            (NP (ADJ superfluous)
                (N Matter)))
    (. .))
```

`Do-sbj`. Finally, in direct questions, auxiliary *do* precedes the subject instead of following it. This inversion occurs in both positive and negative questions, and so retrieving this sentence type relies on the parser correctly identifying the subject via the SBJ function tag. The annotation guidelines for PPCEME require direct questions to be annotated as CP-QUE-MAT immediately dominating IP-SUB. In this context, IP-SUB is understood as part of the direct question rather than an ordinary subordinate clause.

```
(CP-QUE-MAT (WADVP (WADV How))
    (IP-SUB (DOP do's)
            (NP-SBJ (D this) (N Sute))
            (VB fit)
            (NP-OB1 (PRO me)))
    (NP-VOC (NPR Dauy))
    (. ?))
```

G.2 Sentence Types Without Auxiliary *Do*

We now illustrate the historical precursors of the modern sentence types just discussed. In all 3 old forms, it is the main verb (rather than auxiliary *do*) that appears in a past or present tense

[12] We are concerned only with sentences without modal verbs (*can, will*, etc.), aspectual auxiliaries *have* and *be*, or main verb *be*; sentences containing these elements were not affected by the change.

text	# tokens	recall	prec	f1	pos
00-armin-e2	18,768	91.67	92.49	92.08	98.42
01-asch-e1	16,121	89.42	89.96	89.69	98.57
02-bacon-e2	20,181	90.28	91.06	90.67	99.11
03-behn-e3	19,335	92.61	92.69	92.65	99.24
04-blundev-e2	22,619	88.44	90.76	89.58	98.21
05-boethpr-e3	32,806	95.08	95.27	95.17	99.29
06-boylecol-e3	7,544	90.66	91.53	91.09	98.63
07-brinsley-e2	19,710	89.38	89.98	89.68	98.43
08-burnetroc-e3	21,112	93.70	94.05	93.87	99.30
09-clowes-e2	22,500	90.34	91.13	90.73	98.32
10-coverte-e2	20,769	90.83	91.23	91.03	98.39
11-deloney-e2	26,738	93.10	93.43	93.26	98.58
12-elyot-e1	19,157	90.79	91.73	91.26	98.55
13-fabyan-e1	19,029	89.33	89.82	89.57	97.91
14-fisher-e1	10,915	91.13	91.50	91.31	97.27
15-fitzh-e1	18,813	90.51	90.87	90.69	97.74
16-fryer-e3	18,970	88.83	89.29	89.06	97.83
17-gifford-e2	21,148	93.90	94.36	94.13	98.84
18-harman-e1	19,366	90.90	91.80	91.35	98.01
19-hooke-e3	22,494	88.69	88.97	88.83	98.54
20-hooker-a-e2	9,025	91.00	91.79	91.39	98.67
21-hooker-b-e2	8,600	92.13	92.96	92.54	99.09
22-hoole-e3	21,531	89.79	90.27	90.03	98.37
23-jetaylormeas-e3	8,682	93.01	94.03	93.52	99.14
24-jotaylor-e2	31,202	90.72	91.35	91.03	98.50
25-langf-e3	18,351	90.45	90.90	90.67	98.72
26-latimer-e1	17,603	91.40	92.24	91.82	98.53
27-markham-e2	6,138	90.53	91.17	90.85	98.32
28-middlet-e2	19,051	90.09	91.12	90.60	97.25
29-milton-e3	21,307	88.24	89.11	88.67	99.02
30-record-e1	23,422	89.58	89.61	89.59	94.75
31-shakesp-e2	22,032	91.24	91.72	91.48	97.20
32-smith-e2	18,408	94.70	95.07	94.88	99.19
33-stevenso-e1	16,936	84.78	87.10	85.92	93.43
34-stow-e2	17,457	91.66	91.91	91.78	98.75
35-turner-e1	16,302	89.47	90.10	89.78	98.26
36-turnerherb-e1	837	66.55	70.09	68.27	90.20
37-tyndnew-e1	39,476	96.27	96.61	96.44	98.82
38-tyndold-e1	33,901	93.29	93.56	93.42	98.36
39-vanbr-e3	25,052	94.13	94.24	94.18	98.46
40-vicary-e1	19,510	91.36	92.08	91.72	97.89
41-walton-e3	12,557	92.36	92.42	92.39	98.96
total	**805,475**	**91.35**	**91.94**	**91.64**	**98.26**

Table 15: Breakdown of aggregate evalb and POS results for PPCEME overlap files shown in Table 5.

		aligned		non-punc		bullet	
sec	# not aligned	#	acc	#	acc	#	acc
00-armin-e2	277	17,873	88.27	15,858	91.79	6	66.67
01-asch-e1	50	15,960	97.07	13,250	98.13	2	100.00
02-bacon-e2	69	20,140	97.66	17,696	98.81	52	98.08
03-behn-e3	262	19,219	96.72	16,885	98.74	1	0.00
04-blundev-e2	285	22,502	94.24	20,475	95.64	9	88.89
05-boethpr-e3	411	32,765	97.45	29,068	99.07	2	100.00
06-boylecol-e3	17	7,528	97.48	6,793	98.48	0	0.00
07-brinsley-e2	274	19,556	97.21	17,207	98.01	21	85.71
08-burnetroc-e3	44	21,079	97.46	18,832	99.14	0	0.00
09-clowes-e2	108	21,829	95.74	19,360	98.03	36	91.67
10-coverte-e2	40	20,745	95.86	18,268	98.04	87	83.91
11-deloney-e2	436	26,578	95.79	23,553	98.25	4	100.00
12-elyot-e1	630	18,757	96.83	16,789	97.83	2	100.00
13-fabyan-e1	421	18,602	95.86	17,005	97.39	47	82.98
14-fisher-e1	67	10,851	92.43	9,924	96.37	4	75.00
15-fitzh-e1	428	18,640	94.25	16,134	96.88	5	100.00
16-fryer-e3	160	18,933	95.63	16,626	97.46	10	90.00
17-gifford-e2	530	21,112	96.03	18,571	98.51	6	100.00
18-harman-e1	178	17,848	93.32	16,147	96.64	130	79.23
19-hooke-e3	47	22,417	97.06	19,833	98.34	16	100.00
20-hooker-a-e2	45	8,998	97.18	7,950	98.34	2	100.00
21-hooker-b-e2	63	8,578	97.18	7,513	98.70	12	91.67
22-hoole-e3	91	21,436	97.26	19,128	98.11	18	83.33
23-jetaylormeas-e3	175	8,578	89.83	7,697	92.19	27	22.22
24-jotaylor-e2	141	31,074	95.61	27,332	97.74	401	76.81
25-langf-e3	115	18,025	96.48	16,111	98.41	4	100.00
26-latimer-e1	393	17,434	95.21	15,422	97.22	0	0.00
27-markham-e2	79	6,113	86.52	5,579	87.99	5	80.00
28-middlet-e2	2665	18,959	91.80	16,057	95.39	4	75.00
29-milton-e3	33	21,292	97.24	18,470	98.74	6	100.00
30-record-e1	554	23,224	92.24	20,747	93.41	7	57.14
31-shakesp-e2	2259	21,907	91.40	18,220	95.93	5	60.00
32-smith-e2	82	18,381	96.14	16,035	98.88	2	50.00
33-stevenso-e1	763	16,641	88.40	14,569	90.10	385	64.42
34-stow-e2	205	17,422	96.61	15,386	98.39	54	88.89
35-turner-e1	133	16,187	95.72	14,315	97.88	2	50.00
36-turnerherb-e1	7	837	96.30	747	97.46	0	0.00
37-tyndnew-e1	258	39,396	96.13	34,304	98.36	257	93.00
38-tyndold-e1	300	33,780	95.66	30,471	97.52	241	87.55
39-vanbr-e3	3247	24,707	94.22	20,975	97.24	7	57.14
40-vicary-e1	204	19,065	94.36	16,832	97.08	178	85.39
41-walton-e3	697	11736	92.98	10,330	96.56	0	0.00
total	**17,243**	**796,704**	**95.17**	**702,464**	**97.25**	**2,057**	**80.12**

Table 16: Breakdown of aggregate POS results for PPCEME overlap files from Table 5. "aligned" includes all aligned PPCEME tokens (796,704), "non-punc" excludes punctuation tags, and "bullet" includes only words with a bullet character.

tag	gold	EEBO	rec	prec	f1	tag	gold	EEBO	rec	prec	f1
N	93,720	92,513	96.82	95.57	96.19	ADJR	1,530	1527	93.71	93.53	93.62
P	91,175	91,190	98.89	98.91	98.90	EX	1478	1495	97.26	98.38	97.81
,	57,992	71,966	76.91	95.44	85.18	HV	1382	1368	98.98	97.97	98.47
D	62,701	62,440	99.49	99.08	99.29	INTJ	1378	1457	80.44	85.05	82.68
PRO	52,368	52,204	99.34	99.03	99.19	SUCH	1361	1352	99.70	99.04	99.37
CONJ	42,478	42,154	99.44	98.68	99.06	ADJS	1288	1309	95.19	96.74	95.96
ADJ	35,769	35,480	95.93	95.16	95.54	N$	1217	1238	87.96	89.48	88.72
NS	30,937	30,974	96.79	96.91	96.85	ALSO	1212	1205	99.59	99.01	99.30
ADV	24,804	24,477	96.83	95.56	96.19	NPRS	1177	1186	82.21	82.84	82.52
VB	22,724	22,718	97.39	97.37	97.38	BAG	1163	1161	99.40	99.23	99.31
.	36,415	22,274	88.70	54.26	67.33	WD	1127	1145	95.81	97.34	96.57
NPR	19,277	20,210	88.36	92.64	90.45	QS	989	1001	97.70	98.89	98.29
PRO$	17,060	17,023	99.37	99.16	99.26	DOD	887	884	99.66	99.32	99.49
BEP	14,938	14,905	99.14	98.92	99.03	BEN	730	726	99.72	99.18	99.45
VAN	14,540	14,726	95.43	96.65	96.04	DO	728	740	97.16	98.76	97.96
VBP	14,291	14,345	95.88	96.24	96.06	NPR$	707	695	86.19	84.72	85.45
Q	14,044	13,998	98.75	98.43	98.59	OTHERS	487	506	91.50	95.07	93.25
MD	13,828	13,709	99.43	98.58	99.00	HAG	397	393	98.98	97.98	98.48
VBD	13,663	13,653	97.48	97.41	97.44	WARD	320	323	92.57	93.44	93.00
TO	10,890	10,858	99.54	99.25	99.39	WPRO$	315	310	99.03	97.46	98.24
C	9,071	9,113	97.52	97.97	97.75	DAN	266	266	98.12	98.12	98.12
WPRO	7,934	7,920	99.12	98.94	99.03	WQ	258	259	91.51	91.86	91.68
NUM	6,419	6,473	94.72	95.51	95.11	NS$	254	294	67.69	78.35	72.63
VAG	6,181	6,140	95.70	95.07	95.38	FOR	244	259	92.28	97.95	95.03
BED	6,014	6,001	99.60	99.38	99.49	DON	186	184	97.83	96.77	97.30
NEG	5,720	5,691	99.58	99.07	99.33	ADVS	186	150	95.33	76.88	85.12
BE	5,379	5,361	99.22	98.88	99.05	ELSE	133	138	92.03	95.49	93.73
VBN	4,547	4,586	96.14	96.97	96.55	DOI	108	106	94.34	92.59	93.46
FW	4,637	4,332	91.60	85.57	88.48	HVN	89	91	91.21	93.26	92.22
HVP	4,276	4,265	99.11	98.85	98.98	BEI	83	88	81.82	86.75	84.21
RP	4,194	4,182	95.84	95.57	95.70	DAG	55	44	65.91	52.73	58.59
ADVR	3,930	3,880	97.14	95.90	96.52	HAN	52	56	92.86	100.00	96.30
VBI	3,991	3,733	93.65	87.60	90.52	ONES	52	54	94.44	98.08	96.23
WADV	2,874	2,822	98.02	96.24	97.12	NPRS$	36	33	81.82	75.00	78.26
ONE	2,483	2,478	98.95	98.75	98.85	HVI	27	21	85.71	66.67	75.00
OTHER	2,430	2,441	97.91	98.35	98.13	X	24	80	0.00	0.00	0.00
XX	0	2,242	0.00	0.00	0.00	$	23	23	65.22	65.22	65.22
HVD	2,064	2,051	99.07	98.45	98.76	OTHER$	21	22	81.82	85.71	83.72
DOP	2,028	2,026	99.01	98.92	98.96	"	16	9	0.00	0.00	0.00
FP	1,833	1,847	95.34	96.07	95.71	ONE$	12	13	84.62	91.67	88.00
QR	1,718	1,717	98.66	98.60	98.63	’	9	0	0.00	0.00	0.00
OPAREN	1,690	1,703	97.18	97.93	97.55	OTHERS$	6	4	75.00	50.00	60.00
CPAREN	1,661	1,668	97.24	97.65	97.45	NUM$	3	0	0.00	0.00	0.00
total	**796,704**	**796,704**	**95.17**	**95.17**	**95.17**						

Table 17: Complete breakdown by tag of 95.17% score in row “aligned words” in Table 5, extending Table 6. EEBO tags are mapped to PPCEME (gold) tags using token alignment.

form, and it occupies the same position as auxiliary *do*. Thus, we have negative declarative sentences (VERB-DECL-NOT) like:

```
(IP-SUB (NP-SBJ (PRO I))
        (VBD sent)
        (NEG not)
        (PP (P to)
            (NP (PRO you))))
```

negative imperatives (VERB-NOT-IMP) like:

```
(IP-IMP (VBI let)
        (NEG not)
        (IP-INF (NP-SBJ (D that))
                (VB hurt)
                (NP-OB1 (PRO me)))
    (. .))
```

and questions (VERB-SBJ) like:

```
(CP-QUE-MAT
    (WADVP (WADV When))
    (IP-SUB (VBP comes)
            (NP-SBJ (PRO$ your)
                    (N Taylor))
            (ADVP-DIR (ADV hither)))
    (. ?))
```

G.3 Sample CorpusSearch Query

In order to retrieve the 6 diagnostic sentence types, we formulate queries in CorpusSearch (Randall, 2010), a query language for querying, editing, and coding tree structures. Each query is a sequence of boolean conditions on the parser output. For instance, the following query retrieves direct questions with auxiliary *do* (DO-SBJ).

```
    (CP-QUE-MAT* iDoms IP-SUB*)
AND (IP-SUB* iDoms DOD|DOP)
AND (IP-SUB* iDoms NP-SBJ*)
AND (IP-SUB* iDoms DO|VB)
AND (DOD|DOP precedes NP-SBJ*)
AND (NP-SBJ* precedes DO|VB)
```

The asterisks on the labels allow the query to match tokens with further trailing function tags (say, -SPE to indicate direct speech or -RSP for resumptive subjects). Our formulation of the queries assumes that the parser has correctly constructed the relevant clause boundaries.

H Alignment-Mediated Scoring

For the bipartite graph minimum weight matching problem, we use the scipy implementation https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html.

The *PPCEME-gold* trees that are compared against are the original "psd" files from the release, because we wanted to avoid any possibility of preprocessing of PPCEME affecting the "gold" results. These ".psd" trees have empty categories and meta data that affects the spans of the query hits on these gold trees. For example, the lower gold tree in Figure 3(a) has a `(ADVP *T*-1)`, and the span for the VERB-SBJ, `<3272,3278>` includes that empty leaf.

As mentioned in Section 2.2, we follow the preprocessing of the PPCEME files as described in Kulick et al. (2022b,a), for parsing the PPCEME files. This preprocessing removes empty categories and meta data, and so even the AMS evaluation of the *PPCEME-parsed* files requires a token alignment, although a trivial one that simply skips over the empty categories in the alignment from *PPCEME-parsed* to *PPCEME-gold*. Likewise, the alignment used for the AMS scoring of *EEBO-parsed* to *PPCEME-gold* is a slightly modified version of the alignment to *PPCEME-parsed* discussed in Section 2.2, that aligns to the *PPCEME-gold* trees.

I Cross-Validated Results on PPCEME

As briefly discussed in Section 6.2, in addition to the evalb scores on the cross-validated PPCEME sections using the the new language model (Table 3), we also have query-based evaluation (using the method in Kulick et al. (2022b)) for the cross-validation splits. These results are shown in Table 18 and are generally increases over the scores reported in Kulick et al. (2022b).

As discussed in Section 4.1, the parser used on the overlap section is trained on less data than each of these cross-validation splits were trained on, yielding a score of 91.64%, which is lower (as expected) than the cross-validated results using more training data (Table 3). In order to obtain a more robust measure of how the loss in training data affects the parser, we redid the cross-validation split with the training section of each split cut to 55% of its original size. These results are in the bottom half of Table 19, in which, for convenience, we repeat the scores from Table 3. The score of 91.64% is within the expected range.

	DEV				EVAL			
query	# hits	recall	prec	f1	# hits	recall	prec	f1
				Negative declarative sentences				
VERB-NOT-DECL	720	93.73 (3.5)	92.95 (3.9)	93.32 (3.5)	655	94.05 (3.3)	93.68 (2.8)	93.79 (1.4)
DO-NOT-DECL	339	96.53 (2.3)	98.05 (2.7)	97.23 (0.7)	405	96.26 (4.4)	98.58 (2.4)	97.34 (2.2)
				Negative imperative sentences				
VERB-NOT-IMP	120	93.08 (8.0)	90.93 (8.5)	91.69 (6.2)	143	78.16 (14.1)	83.91 (12.8)	80.69 (12.6)
DO-NOT-IMP	41	74.34 (45.9)	71.01 (44.2)	72.55 (44.9)	23	80.0 (38.5)	87.5 (35.4)	82.14 (36.4)
				Questions				
VERB-SBJ	387	89.83 (8.3)	94.65 (4.5)	92.06 (5.9)	190	78.68 (10.2)	87.37 (8.8)	82.18 (5.3)
DO-SBJ	564	92.64 (3.9)	99.01 (1.0)	95.67 (1.9)	329	94.36 (7.1)	99.46 (1.5)	96.75 (4.4)

Table 18: Query-based results for the cross-validation dev and test sections.

	DEV		EVAL	
% train	evalb	POS	evalb	POS
100	92.08 (1.6)	98.23 (0.7)	91.77 (0.6)	98.37 (0.3)
55	91.54 (1.7)	98.02 (0.7)	91.35 (0.7)	98.26 (0.4)

Table 19: Cross-validation parser and POS results. Each result is the mean for the relevant section (dev or test) over the 8 splits (standard deviation in parentheses). Results are reported using both full train section of each split and 55% of the train section.

Phonological processes with intersecting tier alphabets

Daniel Gleim and **Johannes Schneider***
Universität Leipzig
{daniel.gleim,johannes.schneider}@uni-leipzig.de

Abstract

Aksënova and Deshmukh (2018) conjecture that if the phonology of a language requires projection to multiple tiers, the tier alphabets of those tiers are either disjoint or stand in a subset/superset relation, but never form a non-trivial intersection. We provide three counterexamples to this claim.

1 Introduction

An important goal of computational phonology is to determine the complexity of the phonological patterns of natural language. A recent hypothesis is that these patterns are sub-regular and more specifically can be described as tier-based strictly local languages (Heinz et al. 2011, McMullin 2016 i.a.), or slight extensions thereof (Mayer and Major 2018, Graf and Mayer 2018, de Santo and Graf 2019). The general idea is that even non-local processes can be made local over appropriate representations, namely by masking out all irrelevant intervening elements or alternatively, projecting only elements participating in a process on a separate tier where they are adjacent again. Research in that area often focuses on a single pattern/process and a single tier. However, natural languages tend to have more than one phonotactic restriction or more than one phonological process; one might therefore expect that more than one tier is necessary to completely describe the phonology of a language. Moreover, it is the *interaction* of distinct processes that is of particular interest to phonologists. Aksënova and Deshmukh (2018), building on work by McMullin (2016), set out to investigate cases where more than a single tier is needed. They explore the possible relations that the sets of elements on different tiers can stand in: they can be disjoint ($\{a, b\}, \{c, d\}$), they can stand in a subset/superset relation ($\{a, b, c\}, \{b, c\}$) or they can non-trivially intersect ($\{a, b, c\}, \{c, d\}$), i.e. their intersection is neither empty nor the special case of a sub/superset relation (informally *intersection* for the rest of the paper). While being careful to point out the preliminary nature of their work, they claim that no natural language phonology requires a single element to be present on two tiers where each tier contains elements the other does not; in other words that there is no non-empty intersection of tier alphabets that do not stand in a sub/superset relation. They show that, as a function of the number of elements considered, the number of ways to create two sets with a non-empty intersection grows much faster than the respective ways to create true subsets or disjoint sets. As an example, when one considers all possible ways to create proper subsets, disjoint or intersecting sets for 10 elements, the number of intersecting sets already makes up more than 95% of all possibilities. If such a constellation were never to arise, a learner could discard the majority of combinatorially possible multiple TSL grammars. However, in this article we provide three counterexamples to this claim, showing that there are phenomena where one element plays a role in two processes that affect otherwise distinct elements. In Section 2 we provide the necessary background to the use of TSL in phonology and the claims about tier alphabet relationships from Aksënova and Deshmukh (2018). In Section 3, we provide the data for the three counterexamples (Sibe, Tsilhqút'hín, Koryak) that require a description involving overlapping tiers. We close with Section 4 where we discuss an alternative description of two of the three languages as Strictly Piecewise (SP, Rogers et al. 2010); Sibe, however, still resists a description with a single grammar, be it SP or TSL. It remains an open issue whether all existing intersecting TSL phenomena belong to a restricted subset of all possible intersections.

*Authors are listed alphabetically. We thank the participants of the Leipzig phonology reading group for helpful comments and discussion, in particular Sören Tebay for suggesting looking into Koryak.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 243-249.
Amherst, Massachusetts, June 15-17, 2023

2 Background

To familiarize the reader with the TSL-perspective and the type of data Aksënova and Deshmukh (2018) deal with, we provide a short summary of TSL grammars and the examples they give for processes that require disjoint and containing tiers. For more in-depth discussion, the reader is referred to the original paper.

Tier-based strictly local (TSL) grammars (Heinz et al. 2011) work by forbidding substrings of a finite length on a tier. They consist of a tier projection mechanism that scans the original string and projects every segment that is a member of a tier alphabet to a separate tier. There is a set of n-grams of finite size that is forbidden from occurring in the string on the projected tier.

Imagine a toy language with the three vowels *a*, *i* and *u* and an arbitrary consonant inventory. The language requires that all vowels in a word be either high (*i*,*u*) or low (*a*), i.e. we forbid the bigrams **ai,*ia,*au,*ua*. The tier alphabet is the set of all vowels {*a,i,u*}. The projection mechanism projects every vowel from that set it encounters to the tier. A word **blabliblu* thus would have the string *aiu* on its tier. While *iu* is an allowed substring, the combination **ai* is not since it is a forbidden bigram consisting of a high vowel followed by a low one.

Aksënova and Deshmukh (2018) provide an example of processes in a language that require two disjoint tiers, namely vowel harmony and nasal agreement in Kikongo. Vowels have to agree in height; the suffixes *-il/-el* and *-ol/-ul* have a different realization depending on their environment. Examples of different realizations of the former are *-leng-el-* or *-sik-il-*, for the latter *-tomb-ol-* or *-vil-ul-*. In nasal harmony, /*d*/ and /*l*/ become [*n*] if preceded by a nasal in the root, as can be seen for the suffix *-idi*: *-suk-idi-* but *-nik-ini-*. As a result, one needs a nasal harmony tier with the tier alphabet $\{n, m, d, l\}$, forbidding bigrams such as **nd,*nl,*md,*ml*. For vowel harmony, there is a tier with the vowels $\{e, i, o, u\}$, forbidding any bigram with mismatching height features. The tier alphabets of both tiers are disjoint.

(1) $\{n, m, d, l\} \cap \{e, i, o, u\} = \emptyset$

A sub/superset relation is instantiated in Imdlawn Tashlhiyt. Sibilants regressively harmonize in voicing and anteriority. The causative prefix /*s-*/ surfaces as [*s*] in *sːuga*, [*ʃ*] in *ʃfiaʃr* or [*z*] in *zbruzːa*. There are blockers for voicing harmony, namely voiceless obstruents (*sːukz*, not **zːukz*); but they do not act as blockers for anteriority harmony (*ʃquʒːi*, not **squʒːi*). As a result, one needs a tier of all sibilants {*s,z,ʃ,ʒ*}, blocking any bigrams of mismatching anteriority (**ʃʒ*,...), and a second tier for all sibilants and voiceless obstruents {*s,z,ʃ,ʒ,ħ,k,f,χ,q*} to forbid any bigram of adjacent sibilants with distinct values for anteriority (**sz,*ʃz*,...) and forbidding any bigrams of voiced sibilants and voiceless obstruents to model their behaviour as blockers (**zk,*zq*,...). The tier alphabet of the second tier is a superset of the first one.

(2) $\{s,z,ʃ,ʒ\} \subset \{s,z,ʃ,ʒ,ħ,k,f,χ,q\}$

As mentioned above, Aksënova and Deshmukh conjecture that there are no phenomena whose tiers have a non-empty, non-containing intersection. We provide examples of processes that do require intersecting tier alphabets in the next section.

3 Counterexamples

3.1 Sibe

In Sibe (Tungusic, Xinjiang, China), rounding harmony affects all vowels. High back vowels are round if preceded by any round vowel, and non-high vowels agree in rounding with preceding non-high vowels. All the Sibe data is from Li (1996) via Nevins (2005). For the vowel inventory, see Table 1.

	–back		+back	
	–rd	+rd	–rd	+rd
+high	i	y	ɨ	u
–high	ɛ	ø	a	ɔ

Table 1: Sibe vowel inventory

The first effect of rounding harmony is a restriction on a non-round vowel. The high back non-round vowel is not licit following a round vowel (Nevins, 2005: 165):

(3) a. fulxu ‘root’, *fulxɨ
b. ɕøgu ‘vegetable’, *ɕøgɨ

The other high vowels, [*u*] and front high vowels are not restricted in this way and appear freely after vowels with the opposite value for round (Nevins, 2005: 166):

(4) a. χɔnin ‘sheep’
b. narχun ‘slim’

Secondly, non-high vowels must agree in rounding with preceding non-high vowels, as is shown in (5), (Nevins, 2005: 165-167).

(5) a. ɔməl 'grandson', *ɔmɛl, *ɔmal
b. tømχɔ 'nipple', *tømχɛ, *tømχa
c. χɛrχa 'pine tree', *χɛrχø, *χɛrχɔ
d. aχa 'rain', *aχø, *aχɔ

The latter process is restricted to roots, while the former extends to suffixes as well, as can be seen by the examples in (6) and (7).

Following Aksënova and Deshmukh we can establish a vowel tier with all vowels and the conditions on output forms on said tier (Table 2).

Vowel Tier T= {i,y,ɨ,u,ɛ,ø,a,ɔ}	
1.	*[+rd][+high, +back,–rd] H_{r1}\{*yɨ, *uɨ, *øɨ, *ɔɨ \}
2.	*[–high, α rd][–high, $-\alpha$ rd] H_{r2}\{*ɛø, *ɛɔ, *øɛ, *øa, *aø, *aɔ, *ɔɛ, *ɔa \}

Table 2: Tier and Filters for rounding harmony

The second relevant process in Sibe is uvularisation, a long distance vowel-consonant assimilation. Velars in affixes are turned into uvulars if they attach to a root containing a non-high vowel. In (6), no non-high vowel is present in the root, so the affixes surface with a velar. In (7), all root vowels are non-high and the affix consonant is uvularised (Nevins, 2005: 169-170):

(6) Velars with [+high] roots
a. ɕymi(n)-kɨn 'deep-DIM'
b. ulu-kun 'deep-DIM'
c. tyry-xu 'come-PST'
d. tɨ-xɨ 'sit-PST'

(7) Uvulars with [–high] roots
a. ɕa-qɨn 'good-DIM'
b. tɔndɔ-qun 'honest-DIM'
c. gø-χu 'hit-PST'
d. sav-χɨ 'see-PST'

In mixed roots, roots with both high and non-high vowels, the consonant is always uvular, whether it is adjacent to the [–high] vowel or not. Consider (8-b) and (8-d), where the low vowel triggers uvularisation across a high vowel.

(8) Uvulars with mixed roots
a. sul**a**-qɨn 'loose-DIM'
b. χ**ɔ**du-qun 'quick-DIM'
c. tyk**ɛ**-χɨ 'watch-PST'
d. **ø**mi-χɨ 'drink-PST'

The tier that is needed to check uvular assimilation includes velars[1] and [-high] vowels (Table 3).[2] Crucially, it must exclude [+high] vowels since they are transparent. If they were included, they would interfere with the locality on the tier and block uvularisation in mixed roots.

Tier of velars and [–high] vowels T= {k, g, x, ɣ, ɛ, ø, a, ɔ}	
1.	*[–high][+velar] H_{uv}\{*ɛk, *ɛx, *ɛg, *ɛɣ,*øk, *øx, *øg, *øɣ, *ak, *ax, *ag, *aɣ, *ɔk *ɔx, *ɔg, *ɔɣ, \}

Table 3: Tier and filters for uvularisation

We thus have intersecting tiers where [–high] vowels are both in the vowel tier as well as in the uvular assimilation tier but both tiers have elements that are not in the other tier, i.e. velars and [+high] vowels.

(9) $\{i,y,ɨ,u,\mathbf{ɛ,ø,a,ɔ}\} \cap \{\mathbf{ɛ,ø,a,ɔ},k,g,x,ɣ\} \neq \emptyset$

Note that nothing changes about this fact if the vowel tier in Table 2 which handles two processes, rounding harmony for high and for non-high vowels, is split into two: both processes require non-high vowels on their tier, which are crucial for the intersection.

3.2 Tsilhqút'ín

In Tsilhqút'ín (Athabaskan, British Columbia, Canada; all data from Cook 1993, 2013 and Goad 1989), anterior sibilants come in pairs; they have a pharyngealised (or retracted), and a plain version. Anterior sibilants agree long-distance in pharyngealisation. The right-most sibilant functions as the trigger of sibilant harmony and determines the value for every other sibilant. The other sibilants are targets and agree in their retraction value with the rightmost one. Consider (10-a), where

[1] We remain agnostic about which feature distinguishes velar from uvular dorsals.

[2] We also need a third superset tier that includes all vowels and dorsals in order to derive the prohibition on more local [+high][uvular] sequences. Note, though, that it is not possible to describe all three processes on that same tier since high vowels would interfere with the locality of uvularisation and dorsals would interfere in rounding harmony.

the rightmost sibilant is a plain [z] and triggers de-pharyngealisation on the preceding sibilant. (10-b) shows the reverse pattern (Cook, 1993: 160-161):

(10) a. tɛ-**zˤ**-iː-ɬ-**tsæːz** → tɛ**zi**ːɬ**tsæːz**
'I started to cook'
b. næː-**s**ɛ-næː-ɣĩː-l-**tsˤ**ɛ̃**s**ˤ →
na**s**ˤənaːɣə̃ĩl**tsˤ**ə̃**s**ˤ
You are hitting me'

For this process, anterior sibilants must form a tier to the exclusion of everything else (Table 4).[3]

Tier of anterior sibilants T= {s, z, ts, dz, ts', sˤ, zˤ, tsˤ, dzˤ, ts'ˤ}	
1.	*[–R][+R] H_{sib1} {*ssˤ, *szˤ, ... *ts'tsˤ}
2.	*[+R][–R] H_{sib2} {*sˤs, *sˤz, ... *ts'ˤts' }

Table 4: Tier and filters for sibilant harmony

A second non-local process is retraction or 'flattening', where a vowel is retracted[4] in context of a pharyngealised sibilant or a uvular.[5] In (11-a) the uvular [q] triggers retraction of the vowels from /ɛ/ to schwa, and in (11-b) the pharyngealised sibilant acts as the trigger (Goad 1989: 23; Cook 1993: 161):

(11) a. sˤ**ɛ**-l-**q**ʷ**ɛ**s → s**ə**l**q**ʷ**ə**s
'he coughed'
b. χæːtæːs-gɛ̃**s**ˤ → χ**a**ːt**a**ːsˤg**ə̃s**ˤ
'I'll twist it out'

For retraction, therefore, a tier is needed that contains the triggers of the process, pharyngealised sibilants and (labialised) uvulars, and the target, vowels (Table 5).

We thus have intersecting tiers where pharyngealised sibilants are both in the sibilant harmony tier and retraction tier, but the former contains also non-pharyngealised sibilants and the latter vowels

Tier of retraction participants T= {sˤ, zˤ, tsˤ, dzˤ, ts'ˤ, ɢ, ɢʷ, q, qʷ, q', qʷ', χ, χʷ, ʁ, ʁʷ, iː, ɪ, uː, ʊ, æː, ɛ}	
1.	*[–R][+R] R_1 {*iːsˤ, *iːzˤ, ... *ɛʁʷ}
2.	*[+R][–R] R_2 {*sˤiː, *sˤɪ, ... *ʁʷɛ }

Table 5: Tier and filters for retraction

and uvulars.

(12) {s, z, ts, dz, ts', **sˤ, zˤ, tsˤ, dzˤ, ts'ˤ** } ∩
{**sˤ, zˤ, tsˤ, dzˤ, ts'ˤ,** ɢ, ɢʷ, q, qʷ, q', qʷ', χ, χʷ, ʁ, ʁʷ, iː, ɪ, uː, ʊ, æː, ɛ} ≠ ∅

3.3 Koryak

In Koryak (Chukcho-Kamchatkan, Kamchatka, Russia; all data is from Abramovitz 2021) vowels in a word must be from one of three sets. The recessive set {*i, u, e, ə*}, the so-called 'mixed' set {*i, u, a, ə*} or the dominant set {*e, o, a, ə*}. Some vowels are phonetically identical between sets, but need to be distinguished phonologically (for a justification we refer to Abramovitz 2021: ch. 3). A morpheme always has vowels belonging to one set only. If a morpheme with mixed vowels, i.e. a vowel or vowels taken from the 'mixed' set, such as the diminutive *-piɬ* or the root *maqmi* in (13), and a morpheme with recessive vowels are combined, recessive */e/* is lowered to [*a*]. The high vowels and schwa are not affected (Abramovitz, 2021: 60,58):

(13) *e*-lowering
a. ujetiki-piɬ → ujatikpiɬ
'little sled'
b. maqmi-te → maqmita
'with a bow'

If a morpheme with a dominant vowel and a morpheme with a recessive or mixed vowel are combined, recessive and mixed */i/* and */u/* are lowered to [*e*] and [*o*] respectively, and recessive */e/* is lowered to [*a*]. Nothing happens to (mixed or recessive) *a* or schwa. Consider (14), where the same mixed and recessive morphemes as in (13) are now put in a context with dominant vowels (Abramovitz, 2021: 61f):

(14) general lowering
a. ujetiki-piɬɬaq-ŋqo →

[3] We use the more abstract feature R for both sibilant harmony and retraction, as is usual in the literature on Tsilhqút'ín.

[4] The featural changes a vowel partakes in under retraction are complex but irrelevant for this discussion.

[5] This is a gross simplification of the process. There are differences regarding the trigger – sibilant induced retraction is more long-distance than uvular induced retraction – and regarding directionality: leftward retraction is unblockable, whereas rightward retraction may be blocked by velars and long vowels function as icy targets. None of this affects the intersection that we discuss here. For a thorough discussion of the data and theoretical implications we refer to Goad (1989); Mullin (2011); Gleim (2021).

ojatekpeʎʎaqəŋqo
'from the small sled'

b. qoja-te → qojata
'by reindeer'

Vowel harmony in Koryak is obviously less phonetically grounded than the processes discussed above. We will implement it in a TSL grammar with diacritic features instead of the more usual phonological ones and leave any discussion of naturalness aside.

We will assume the diacritic features R, M and D which are part of the vowels' specifications that derive these classes. This gives us the vowel inventory in (15). To reduce clutter, recessive vowels do not carry diacritics.

(15) {e, i, u, ə, a^M, i^M, u^M, $ə^M$, e^D, $ə^D$, a^D, o^D }

First, we will present a convenient tier for each process individually and show that the tiers do intersect. After that, we show that the tiers can neither be reconstructed as a single tier nor as tiers in a superset-subset relation. The tier that derives *e*-lowering (Table 6) must contain all vowels with the M-diacritic as well as recessive *e*.

Tier of e and M-vowels T= {e, i^M, u^M, a^M, $ə^M$}	
1.	*Me, *eM eL\{*ei^M, *eu^M, *ea^M, *$eə^M$, *i^Me, *u^Me, *a^Me, *$ə^M$e }

Table 6: Tier and filters for e-lowering

On the tier that derives general lowering (Table 7), all dominant vowels, all recessive vowels except recessive schwa, and *i* and *u* with the M-diacritic must be present, but crucially not *a* with the M-diacritic.

Both tiers share *e* and the high vowels with the M-diacritic, but only the first contains the non-high vowels with the M-diacritic; and only the second the recessive high vowels and the dominant vowels:

(16) {a^M, $ə^M$, **e, i^M, u^M**} ∩ {**e, i^M, u^M**, i, u, o^D, a^D, e^D } ≠ ∅

Now, let us consider alternatives with non-intersecting tiers. If we conflate the two tiers above into a single tier, which contains every vowel but recessive schwa, we run into problems with a sequence like the one in (17).

Tier of dominant vowels, high vowels and *e* T= {e, i, u, i^M, u^M, e^D, $ə^D$, a^D, o^D}	
1.	*DR, *RD GL_1\{*o^De, *o^Di, *o^Du, *a^De, *a^Di, *a^Du, *e^De, *e^Di, *e^Du, *$ə^D$e, *$ə^D$i, *$ə^D$u, *eo^D, *io^D, *uo^D, *ea^D, *ia^D, *ua^D, *ee^D, *ie^D, *ue^D, *e$ə^D$, *i$ə^D$, *u$ə^D$}
2.	*DM, *MD GL_2\{ *$o^D$$i^M$, *$o^D$$u^M$, *$a^D$$i^M$, *$a^D$$u^M$, *$e^D$$i^M$, *$e^D$$u^M$, *$ə^D$$i^M$, *$ə^D$$u^M$, *$i^M$$o^D$, *$u^M$$o^D$, *$i^M$$a^D$, *$u^M$$a^D$, *$i^M$$e^D$, *$u^M$$e^D$, *$i^M$$ə^D$, *$u^M$$ə^D$}

Table 7: Tier and filters for general lowering

(17) *e-i-i^M*

e-i and *i-i^M* are both perfectly fine bigrams, so the structure as a whole should be fine. However, in Koryak we actually get an output with a lowered */e/* in such a configuration. The second alternative for making the Koryak tiers compatible with Aksënova and Deshmukh's hypothesis, is to project the elements that are uniquely in the *e*-lowering tier, a^M and $ə^M$, into the general lowering tier as well. This yields unwanted results in strings like (18).

(18) i-a^M-o^D

Again, both *i-a^M* and *a^M-o^D* are perfectly fine sequences on the general lowering tier. Only by banning a^M from the tier, we get the desired violation of **io^D*. To conclude, the tiers we proposed for each process individually do derive the data correctly and are necessarily intersecting.

4 Discussion

The absence of phonological processes that share a subset of their elements would have been computationally appealing since it would have eliminated a large share of logically possible tier alphabet relations. However, as we have demonstrated above, such processes do in fact exist. This raises the question if tiers of two interacting processes can form any possible intersection of their tier alphabets or if these intersections are subject to additional restrictions that at least somewhat narrow down the combinatorial possibilities.

One such possibility would be that all phenomena that require intersecting tiers in a multiple TSL description can be described by a single grammar from a class that is incomparable to TSL, i.e. a

class that neither contains nor is contained by TSL. We want to mention one such class that has previously been used in the literature, Strictly Piecewise (Rogers et al. 2010), that works for two of the processes but unfortunately fails for Sibe. Strictly Piecewise (SP) is a class that is incomparable to TSL (see e.g. de Santo and Graf 2019 for an overview of containment relationships of classes). Due to the 'global' nature of vowel harmony in Koryak, it is possible to describe both phenomena discussed in 3.3 with a single SP grammar. Intuitively, Strictly Piecewise grammars forbid certain subsequences of strings, regardless of the number and nature of intervening elements. As an example, we can forbid that a dominant vowel is followed by a recessive vowel at any distance in a word by forbidding e.g. the subsequence $*o^D e$ (and the reverse for the equally forbidden co-occurrence). With this, one can simply list all impossible co-occurrences of vowels from different classes without worrying about interveners. As far as we can see, this derives the Koryak data just like the tier-based procedure above. The same goes for the (simplified version of) the Tsilhqút'ín data. As already mentioned in Rogers et al. (2010), sibilant harmony can be modelled as SP by forbidding subsequences of mismatching sibilants. This derives the process described by our first tier. To add vowel retraction in the context of pharyngealised sibilants and uvulars does not interfere with the first process; we can state further co-occurrence restrictions for vowel-sibilant/uvular combinations in the same SP grammar. The joint statement of such restrictions is not possible in a unified tier-based attempt where the additional vowels would interfere with the locality on the tier for sibilant harmony.

A potential conjecture that all processes that require intersecting tiers can be described by a single SP grammar unfortunately fails for Sibe: we know that the next vowel after [*y*] cannot be [*ɨ*] (**yɨ*), yet (8-c) *tykɛ-χɨ* is well-formed. This is because neither *yɛ* nor *ɛɨ* are problematic vowel sequences due to the opaque nature of *ɛ*. A simple SP-grammar cannot describe such blocking effects. One needs to simultaneously rule out **yɨ* and rule in *yɛɨ* subsequences. SP cannot distinguish both cases. Another option are classes that use more fine-grained projection mechanisms for their tiers such as input (and/or) output-TSL, I/O-TSL (de Santo and Graf 2019, Mayer and Major 2018, Graf and Mayer 2018). Intuitively, one can specify that a certain segment is only projected if it is preceded/followed by another specific segment in the input string (ITSL); or that a segment is only projected if it then precedes/follows a specific segment on the tier (OTSL); or a combination of both (IO-TSL). A reviewer asks whether a single IO-TSL can be used to describe the Sibe data. One option would be to project all vowels, but only project velars if they are then preceded at some distance by a [-high] vowel on the already existing tier. However, we have seen in (8) that the relation between the relevant dorsals and [-high] vowels is non-local. Whether a finite distance between a [-high] vowel and a dorsal is possible depends on whether recursive word formation processes (e.g. repeated affixation) are attested. We follow the practice of treating non-local processes as unbounded if they are only constrained by the maximal size of existing words (as is implicit e.g. in the treatment of the data from Aksënova and Deshmukh 2018).

Therefore it remains to be seen if the phonologies of natural languages allow all possible tier alphabet intersections or if there are hidden restrictions such that all intersecting tier alphabets can be described by a single class of languages incomparable to TSL.

References

Rafael Meghani Abramovitz. 2021. *Topics in the grammar of Koryak*. Ph.D. thesis, MIT.

Alëna Aksënova and Sanket Deshmukh. 2018. Formal Restrictions On Multiple Tiers. In *Proceedings of the Society for Computation in Linguistics*, volume 1, pages 64–73. University of Massachusetts Amherst.

Eung-Do Cook. 1993. Chilcotin flattening and autosegmental phonology. *Lingua*, 91(2-3):149–174.

Eung-Do Cook. 2013. *A Tsilhqút'ín grammar*. UBC Press, Vancouver, BC.

Aniello de Santo and Thomas Graf. 2019. Structure Sensitive Tier Projection: Applications and Formal Properties. In *Formal Grammar*, pages 35–50. Springer Berlin Heidelberg.

Daniel Gleim. 2021. Theoretical Implications of Directionally Asymmetric Transparency. *Proceedings of the Annual Meetings on Phonology*, 9.

Heather Goad. 1989. On the feature [rtr] in Chilcotin: A problem for the feature hierarchy. Ms., University of Arizona, Tucson.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-Retroflexion is Input-Output Tier-Based Strictly Local. In *Proceedings of the Fifteenth Workshop on*

Computational Research in Phonetics, Phonology, and Morphology, pages 151–160, Brussels, Belgium. Association for Computational Linguistics.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based Strictly Local Constraints for Phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 58–64, Portland, Oregon, USA. Association for Computational Linguistics.

Bing Li. 1996. *Tungusic vowel harmony. Description and Analysis*. Ph.D. thesis, University of Amsterdam.

Connor Mayer and Travis Major. 2018. A Challenge for Tier-Based Strict Locality from Uyghur Backness Harmony. In *Formal Grammar 2018*, pages 62–83, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kevin James McMullin. 2016. *Tier-based locality in long-distance phonotactics : learnability and typology*. Ph.D. thesis.

Kevin Mullin. 2011. Strength in harmony systems: Trigger and directional asymmetries. Ms., University of Massachusetts Amherst.

Andrew Nevins. 2005. *Conditions on (Dis)Harmony*. Ph.D. thesis, MIT.

James Rogers, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome, and Sean Wibel. 2010. On Languages Piecewise Testable in the Strict Sense. In Christian Ebert, Gerhard Jäger, and Jens Michaelis, editors, *Lecture Notes in Computer Science*, pages 255–265. Springer Berlin Heidelberg.

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 [$_{DO}$ the regulations] to [$_{IO}$ Jim].
b. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the regulations regarding import and export taxes for pottery].
c. ? Emma explained to [$_{IO}$ Jim] [$_{DO}$ 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

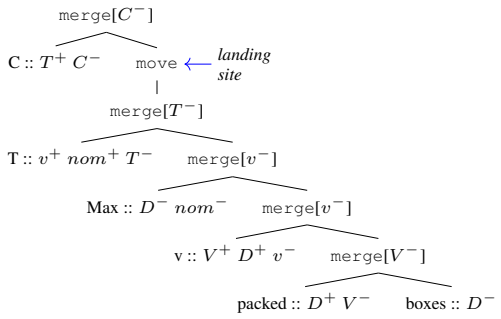
On an intuitive level, the MG parsing model used in this study infers processing difficulties of a given sentence according to how memory-costly it is to parse by a parser for MG.

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)



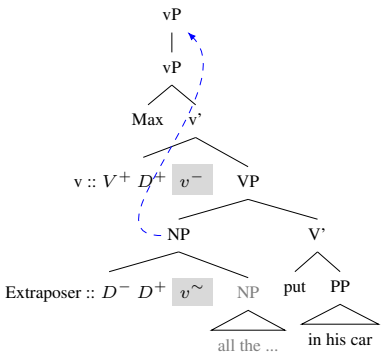
In (3), the uppercase features $X^{\pm}$ are `merge` features. The superscripts $^+$ and $^-$ indicate selector and category for merge, respectively. For instance, in the bottom of the tree, *packed* :: D^+ V^- merges with *boxes* :: D^- and "checks" the matching D feature. The lowercase features $y^{\pm}$ 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)

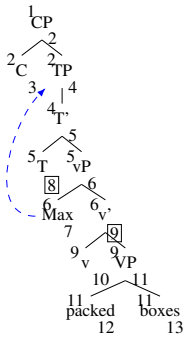


Despite its now unpopularity due to alternative proposals such as Kayne's 1994 Linear Correspondence Axiom (LCA), rightward movement is not discarded by the MGs on formal grounds. More broadly, this shows how closely MGs can incorporate "devices from the syntacticians' toolbox" (Graf et al., 2017), which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

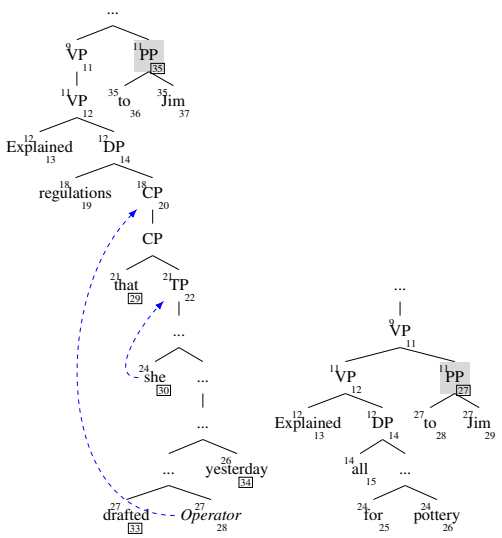
(7) a. Emma explained [all the regulations

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 [$_{DP}$ boxes] [$_{PP}$ in a car]. (short-DP short-PP)

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

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

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

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

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

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

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

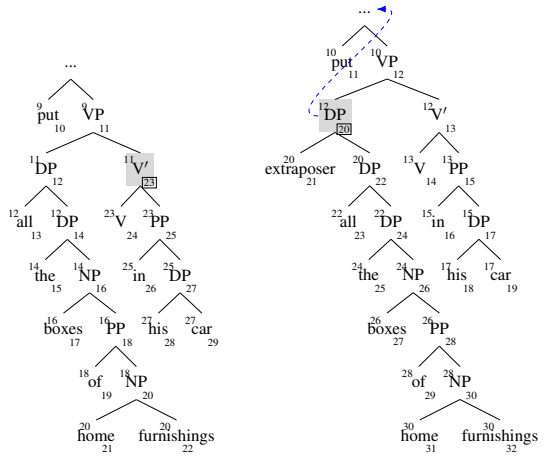
Given the behavior data, we expect the parser to predict an object shift advantage *only* for the pair in (10) ((10b) is advantageous). The pair in (8) contains no heavy constructions, thus no shift advantage is expected. The pair in (9) contains heavy PPs, but the canonical order is the end-weight order, 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.

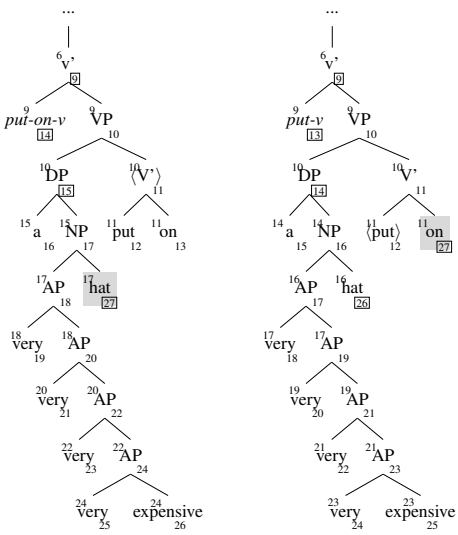
Assuming a particle stranding analysis for separated PVs, and a complex verb raising one for the

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.

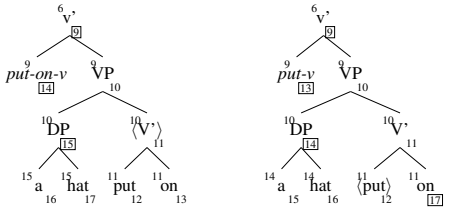


(a) **put on** a very very...hat. (b) **put** a very very...hat **on**.

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).

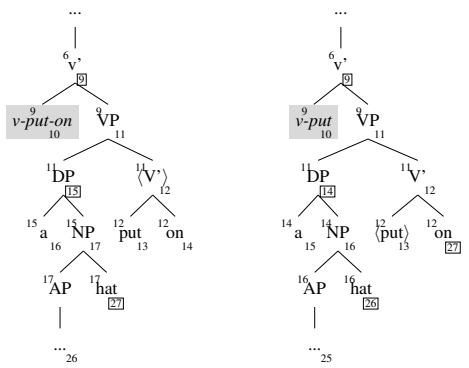


(a) **put on** a hat. (b) **put** a hat **on**.

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).

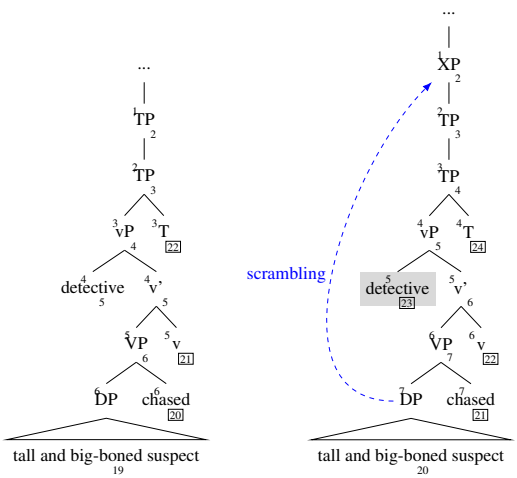
(15) a. [$_O$ Se-ga takakute gassiri sita
height-nom tall-and big-boned
hanni-o]$_i$ [$_S$ Keezi-ga] t_i Oikaketa.
suspect-acc detective-nom chased
'The detective chased the suspect who is tall and big-boned.'

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.

Rethinking representations: A log-bilinear model of phonotactics

Huteng Dai
Department of Linguistics
Rutgers University
huteng.dai@rutgers.edu

Connor Mayer
Department of Language Science
University of California, Irvine
cjmayer@uci.edu

Richard Futrell
Department of Language Science
University of California, Irvine
rfutrell@uci.edu

Abstract

Models of phonotactics include subsegmental representations in order to generalize to unattested sequences. These representations can be encoded in at least two ways: as discrete, phonetically-based features, or as continuous, distribution-based representations induced from the statistical patterning of sounds. Because phonological theory typically assumes that representations are discrete, past work has reduced continuous representations to discrete ones, which eliminates potentially relevant information. In this paper we present a model of phonotactics that can use continuous representations directly, and show that this approach yields competitive performance on modeling experimental judgments of English sonority sequencing. The proposed model broadens the space of possible phonotactic models by removing requirements for discrete features, and is a step towards an integrated picture of phonotactic learning based on distributional statistics and continuous representations.

1 Introduction

Phonotactics refers to restrictions on how sounds can be sequenced in a language. For example, although neither *blick* [blɪk] nor *bnick* [bnɪk] are real English words, native speakers feel that *blick* could be an English word, while *bnick* could not because it begins with the prohibited onset *[bn] (Chomsky and Halle, 1965). Phonotactic restrictions vary between languages, meaning that they must be learned. For example, *steek* [stik] is a possible word in English but not in Spanish, because the latter has a phonotactic restriction on syllables beginning with [st]. As learners acquire a language, they become sensitive to the frequencies of different sequences. This phonotactic knowledge underlies speakers' intuitions about possible words in their language.

Experimental studies involving acceptability judgments have found that speakers have **gradient intuitions** about phonotactic well-formedness (e.g., Coleman and Pierrehumbert, 1997; Albright, 2009; Hayes et al., 2009; Daland et al., 2011). For example, when considering the nonce words *blick* [blɪk], *bnick* [bnɪk], and *bwick* [bwɪk], English speakers typically find *blick* to be acceptable, *bnick* to be poor, and *bwick* to be intermediate between the two (Albright, 2009). This has led to the development of **probabilistic** models of phonotactics, which assign a continuous score to words that reflects their gradient well-formedness (Hayes and Wilson, 2008; Futrell et al., 2017; Wilson and Gallagher, 2018; Gouskova and Gallagher, 2020; Mayer and Nelson, 2020). Phonotactics is also commonly treated as probabilistic in models of higher-level linguistic tasks, such as speech perception and word segmentation (see discussion in Daland, 2015).

1.1 Feature-based generalizations

An additional difficulty for phonotactic models is the problem of **accidental gaps**: sequences of sounds that do not appear in the lexicon but are judged to be acceptable. Humans do not treat unattested sequences uniformly: in the example in the previous section, both [bw] and [bn] are unattested onsets in English, but the former is preferred to the latter. Phonotactic models thus need to be able to generalize to unseen sequences in a way that is consistent with human behavior.

The standard solution is to have models operate on **featural representations**, which decompose segments into sets of feature-value pairs (or, alternatively, a vector of values whose dimensions are the features). Features allow models to refer to classes of segments based on shared properties. In English, for example, the feature vector [−continuant] characterizes the set of stops and affricates, [−sonorant] picks out the set of obstru-

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 259-268.
Amherst, Massachusetts, June 15-17, 2023

ents, and [−continuant, −sonorant] picks out the set of obstruent stops/affricates (excluding the nasal stops). Returning to the example above, although [bw] and [bn] are both unattested onsets, there are many onsets that are featurally similar to [bw], consisting of b[+approximant] sequences like [bj], [bl], [bɹ]. There are none that are similar to [bn], consisting of b[−continuant]. Features allow these kinds of generalizations to be modeled.

1.2 Whence features?

Phonological features are typically defined with respect to phonetic properties (e.g., Chomsky and Halle, 1965). This reflects the strong typological tendency that sounds with similar phonetic properties tend to pattern similarly.

More recent research has proposed that features may be **emergent**, reflecting shared, language-specific distributional properties in addition to phonetic properties (e.g., Mielke, 2008; Archangeli and Pulleyblank, 2018; Gallagher, 2019; Archangeli and Pulleyblank, 2022). There are several motivations for this perspective.

First, a central desideratum in designing feature systems is to allow them to reference all and only the classes of sounds that pattern together cross-linguistically: namely, those that share some subset of phonetic properties encoded by the feature system. However, linguists have discovered a substantial number of phonological classes across languages that cannot be referenced under standard feature systems (Mielke, 2008). An example of one such class is the segments that participate in a nasalization process in Evenki (Tungusic; Nedjalkov, 1997; Mielke, 2008): the sounds /v s g/ become nasalized following a nasal consonant, but similar sounds such as /b d x/ do not. It is not possible to provide a set of feature/value pairs that picks out the class /v s g/ to the exclusion of all other sounds in the language, which predicts that it should not pattern cohesively. In similar cases, researchers have proposed modifications to existing feature systems to account for unexpected classes (though perhaps not modifications so extreme as to capture /v s g/; e.g., Rice and Avery, 1989; McCarthy, 1991; Paradis and LaCharité, 2001).

Emergent feature theory instead proposes that features may be learned in part from the distributional patterning of sounds, which means a shared representation could be learned for irregular classes like /v s g/ if the language data supported it. This also turns the focus away from enumerating all of the features motivated by natural language phonology, focusing instead on how features might be learned from the phonetic and distributional properties of sounds.

A second, related, motivation for emergent features is the variable patterning of the same segment across different languages. For example, Mielke (2008) notes that some languages treat /l/ as [+continuant], and others treat it as [−continuant]. Both are sensible from the perspective of phonetic substance, since /l/ is [-continuant] mid-sagittally but [+continuant] off mid-line. Rather than trying to determine the "correct" value of [continuant] for /l/, or perhaps to split [continuant] into a pair of features corresponding to on and off the mid-line, emergent feature theory suggests that the featural representation of /l/ can vary depending on whether it patterns with [+continuant] or [−continuant] sounds in a language.

Several computational models have been proposed to test the plausibility of distributional learning of phonological classes/features (e.g., Goldsmith and Xanthos, 2009; Mayer, 2020; Nelson, 2022). These papers have tested phonological class learning under the extreme assumption that the learner has no access to the substantive phonetic properties of segments, but only their statistical patterning. Representations learned from distribution alone have been shown to capture non-trivial phonetic distinctions as well as distribution-specific information (Goldsmith and Xanthos, 2009; Mayer, 2020) and to perform comparably to phonetic features in downstream tasks (Nelson, 2022).

The segmental representations in such models are learned using similar techniques to distributional word embeddings (Mikolov et al., 2013; Levy and Goldberg, 2014), which produce real-valued vector representations. In phonological theory, features serve as an extensional description of phonological classes, and most models of phonotactics assume discrete features accordingly. A common feature of the models above is that they use clustering techniques to convert these continuous representations into discrete classes. These classes can then be converted into discrete featural representations (Mayer and Daland, 2020).

Although the process of converting continuous representations to discrete ones aligns with the standard theoretical treatment, it discards information and introduces additional degrees of freedom into

the learning process, in the sense that choices must be made about how clustering is done and how features are derived from classes. Several neural models of phonotactics have used continuous representations directly (Mirea and Bicknell, 2019; Mayer and Nelson, 2020). These recurrent neural network models perform well but are difficult to interpret in a theoretically-satisfying way because they involve many nonlinear transformations of the input features.

1.3 Overview of this paper

This paper presents a computational model[1] that bridges the gap between distributional learning techniques and phonotactic models by incorporating the induction of continuous distributional representations into the overall framework of phonotactic learning. More specifically, we will show that (a) the proposed model is flexible enough to make use of a range of different featural representations, including the continuous features typically produced by distributional learning techniques; (b) the model performs comparably to other models in the field; and (c) the continuous distributional representations result in better generalization to new data than their discretized counterparts, and outperform phonetic features in some respects.

Sections 2 and 3 describe the proposed model and three types of featural representation that will be used to test the model. Section 4 presents a simple toy example to illustrate the performance of the model, and Sections 5 and 6 compare the performance of the model on English onsets against several other models of phonotactic learning. Section 7 offers a brief discussion.

2 Model description

Our goal is to develop a model for the probability of a form in terms of the conditional probability of a symbol x given its preceding context c, in a way that leverages potentially real-valued featural representations of x and c, such as those resulting from distributional analysis, without needing to reduce these continuous representations into hard categories or clusters. To these ends, we adopt **log-bilinear** probability models, a generalization of the widely used log-linear model. Below, we first describe log-linear models and their relation to existing models of phonotactics, then their generalization to log-bilinear models.

2.1 Log-linear models

In a log-linear model, a form is assigned a probability as a function of weighted features.[2] One example is the Maximum Entropy phonotactic model proposed by Hayes and Wilson (2008), in which a wordform x is described in terms of a constraint violation profile: a vector $\phi(x)$ whose values are the number of times the wordform violates each constraint. The probability of x under the model is then

$$p(x) \propto \exp\Big\{\mathbf{w}^\top \phi(x)\Big\}, \tag{1}$$

where the weight vector $\mathbf{w}$ represents the weight of each constraint. The vector $\mathbf{w}$ is found by optimization to maximize the likelihood of a given dataset of forms.

Such models are called *log-linear* because the function in Eq. 1 is linear after taking a logarithm. In the context of phonotactics, the linear component of this model is a Harmonic Grammar model (Smolensky and Legendre, 2006; Pater, 2009) that uses numerical constraint weights and assigns each word a numerical score based on its violation profile. Log-linear models are one way of using these scores to compute a probability distribution over words (cf. Boersma and Pater, 2016).

Log-linear models are ubiquitous not only in computational learning models but also in natural language processing (e.g., Berger et al., 1996; Della Pietra et al., 1997). Before the modern renaissance of neural networks, the dominant paradigm for any supervised classification task in NLP (for example, the task of reading in a movie review and then outputting the probability that the review is positive) was to use a hand-crafted featural representation $\phi(x)$ of the text input x and to learn optimized weights $\mathbf{w}$ to maximize the likelihood of labels in training data (Jurafsky and Martin, 2023).

2.2 The current proposal: log-bilinear model

The log-bilinear model extends the log-linear model to make the weights conditional on the features of the context. Instead of finding an optimal weight vector, in a log-bilinear model one finds an optimal weight *matrix* that relates the representations of the context to the representations of the outcome. Such models were initially developed in a

[1] The code and data used in this paper can be found at https://github.com/hutengdai/vector_bilinear.

[2] Features in this context refer to properties of the form in general, not necessarily phonological features.

language modeling context to predict words given previous words (Mnih and Hinton, 2007, 2008; Mikolov et al., 2013; Futrell, 2022).

We apply a log-bilinear model in the setting of calculating the conditional probability of an individual segment x conditional on a context c, given vector representations of the segment $\phi(x) \in \mathbb{R}^K$ and of the context $\psi(x) \in \mathbb{R}^L$. The model is defined as

$$p(x \mid c) \propto \exp\Big\{\psi(c)^\top \mathbf{A}\phi(x)\Big\}, \qquad (2)$$

where $\mathbf{A} \in \mathbb{R}^{K\times L}$ is an **interaction matrix** that defines how the features of the context $\psi(c)$ relate to the features of the result $\phi(x)$. The entry A_{kl} in the interaction matrix is an association weight for the kth feature of the context and the lth feature of the next segment; a high value of A_{kl} means (all else being equal) that a segment with a high value of the lth feature is likely to follow in a context with a high value of the kth feature.

The interaction matrix $\mathbf{A}$ is found to maximize the likelihood of a training dataset consisting of N context–outcome pairs $\{\langle c_n, x_n\rangle\}_{n=1}^N$:

$$\mathbf{A} = \arg\max_{\mathbf{A}} \sum_{n=1}^{N} \log p(x_n \mid c_n). \qquad (3)$$

The implemented learning algorithm discovers the interaction matrix using the Adam optimization algorithm (Kingma and Ba, 2015), starting from a randomly-initialized $\mathbf{A}$ whose entries are all drawn from a standard Normal distribution.

We model the likelihood of a wordform in terms of features of segmental bigrams. That is, the weights learned by the model correspond to the strength of bigram constraints on the features of two adjacent segments. The probability for a form $\sigma_1, \ldots, \sigma_T$ is then

$$p(\sigma_1, \ldots, \sigma_T) = \prod_{t=1}^{T} p(\sigma_t \mid \sigma_{t-1}), \qquad (4)$$

where $p(\cdot \mid \cdot)$ is a log-bilinear model with the same featurization $\phi(\cdot)$ for the current segment σ_t and the context σ_{t-1}. This restriction to featural bigram constraints is an implementation detail; the log-bilinear model works with any vector representation of context and target. In particular, context and target representations do not need to be the same size; the context representation can include information about multiple segments by increasing the dimension of $\psi(c)$ and $\mathbf{A}$ accordingly.

3 Featurizations

We will illustrate the performance of the log-bilinear model described above using three types of featurizations that have been used in the literature on phonotactic learning: **discrete phonetic** features, **continuous distributional** features, and **discrete distributional** features. The purpose of these comparisons is to (a) demonstrate the flexibility of the model in terms of representational choices; and (b) show that the continuous distributional representations contain useful, fine-grained information that is lost when these representations are discretized.

3.1 Discrete phonetic features

An obvious choice for the featurization of a segment σ is the discrete phonetic features that are commonly used in phonological theory. We adopt the featurization system from Hayes (2009).

For models where featural representations are treated as numerical vectors, such as the log-bilinear model, we adopt a **binary featurization** that identifies each dimension of $\phi(\sigma)$ with a phonological feature *and its possible values*. So for example, there would be a separate dimension for the feature-value pairs $[+\text{continuous}]$ and $[-\text{continuous}]$ with value 1 if that feature-value pair applies to the segment σ and 0 otherwise. For example, the segment [k] would receive the vector representation

$$\phi(\text{k}) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \begin{matrix} +\text{dorsal} \\ -\text{dorsal} \\ +\text{continuous} \\ -\text{continuous} \\ +\text{consonantal} \\ -\text{consonantal} \\ \vdots \end{matrix}. \qquad (5)$$

This leads to a more expressive featurization than encoding negative values as -1. This would force the effect of a negative feature value to be the inverse of the effect of a positive feature value, whereas the binary featurization allows positive and negative values to have independent effects.

3.2 Continuous distributional representations

We induce continuous representations based on their statistical distributions in the training data by calculating probabilities of segments in different contexts and then converting these into

Pointwise Mutual Information (PMI; Church and Hanks, 1990). PMI is an information-theoretic measurement that compares the joint probability of two events against the product of their individual probabilities. PMI and the related Positive PMI have been used in previous models of distributional phonotactic learning (Silfverberg et al., 2018; Mayer, 2020; Nelson, 2022).

PMI is defined as follows:

$$\text{PMI}(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)}. \quad (6)$$

If $p(x)$ and $p(y)$ are independent this value will be close to zero, while if they occur together more/less frequently than chance, it will be positive/negative. Here we define $p(x, y)$ to be the joint probability of segment x followed by segment y. We compute the probabilities using a bigram language model with Kneser-Ney smoothing (Chen and Goodman, 1999), implemented using the `lm` module from the `nltk` Python library (Bird et al., 2009). This model produces conditional probabilities of the form $p(y|x)$, which we convert to joint probabilities $p(x, y)$.

The dimensions of these representations are the PMI values of the segment in each context in the training data. Following Mayer (2020), we consider both preceding and following context by running a pair of language models: one that runs forward to calculate PMI values based on preceding context, and one that runs backwards to calculate PMI values based on following context. These two vectors are concatenated to produce the full representation.

3.3 Discrete distributional features

We also include discrete distributional featurizations derived from the continuous representations in the previous section. This discretization step allows the distributional representations to be used in models that assume discrete features.

Converting continuous features to discrete ones involves two steps: a *clustering step* where classes of segments are identified based on similarities in their continuous representations, and a *feature assignment step* where a feature system is derived from these classes.

We include two clustering strategies: the recursive clustering algorithm described in Mayer (2020) and the SC COV algorithm from Nelson (2022). Both of these involve using the continuous embeddings to compute graph structures that reflect distributional similarity between segments, and then applying graph partitioning techniques to derive classes of segments. For reasons of space we refer the reader to the respective papers.

We follow Nelson (2022) in using the *inferential complementary* algorithm from Mayer and Daland (2020) for feature assignment. Mayer and Daland (2020) presents a suite of algorithms that derive a feature system from a set of input classes based on subset/superset relationships between them, differing in what values are permitted and whether complement classes of the input classes are inferred. The inferential complementary algorithm adds complement classes of the input classes with respect to their parents and assigns both $+$ and $-$ feature values.[3]

4 A toy example of the log-bilinear model

We present a simple toy example below to illustrate the performance of the log-bilinear model using the continuous distributional features described in Section 3.2. We define a language over the alphabet {C, V, #}, where # is a word boundary. The language has a restriction on adjacent CC sequences, and the training data is {VCV, CVC, CVV, VVC, VVV} (word boundaries are omitted for clarity). The continuous distributional featurization of each segment calculated from the training data is shown in Table 1. Sequences that are unattested in the training data, such as ## or CC, have large negative scores, while more commonly observed contexts have positive scores.

	#_	C_	V_
#	-2.492	0.504	0.232
C	0.517	-3.256	0.251
V	0.111	0.118	-0.278

Table 1: Continuous distributional representations of the segments in the toy language. Each row is a representation of a segment, and the columns are the PMI values of that segment in the context indicated by the column label. For simplicity's sake we only present preceding contexts here, but the full model also includes dimensions corresponding to following context.

[3] Nelson (2022) in fact uses a slightly simplified version of this algorithm: the original algorithm recursively adds complement classes until there are no more to add, while the algorithm in Nelson (2022) adds complement classes once and then terminates. This potentially reduces the expressivity of the feature system, but the two approaches are similar enough that we treat them as a single feature assignment strategy.

Table 2 shows the scores assigned by the log-bilinear model to a set of nonce words after it was fitted to the training data using the representations in Table 1. The model successfully assigns a lower probability to words containing a CC sequence.

Word	Score
C V C V	5.397
V C V V	5.980
V V V V	6.393
C C C V	8.825
V C C C	8.933
C C C C	10.272

Table 2: Scores assigned by the trained model to nonce forms. The scores here are negative log probabilities.

5 Model comparison

We evaluate the performance of the log-bilinear model against several existing models of phonotactics. These models take as input a set of training data and, in most cases, a set of featural representations for the segments in the training data. Fitted models assign scores to word forms that reflect their probabilities.

The purpose of this comparison is to demonstrate that the log-bilinear model performs favorably against existing phonotactic models.

5.1 Hayes and Wilson learner

The Hayes and Wilson learner (Hayes and Wilson, 2008) is a Maximum Entropy model of phonotactics. We refer the reader back to Section 2.1 for a description of how word probabilities are computed based on input constraint violation profiles and a set of learned weights.

In addition to fitting weights, the Hayes and Wilson learner also simultaneously learns the constraints themselves from the data, up to an upper bound specified by the user. Constraints are implemented as featural n-gram constraints (e.g., *[−voi, −son][+voi, −son]). Constraints are discovered by comparing observed vs. expected counts in the training data and selecting constraints that penalize structures with unexpectedly low counts. There is a bias towards constraints that include fewer features, but more complex interactions are learned when the data support them.

The scores assigned by this model are *harmony values*, which are unnormalized log probabilities (the linear component of the log-linear model).

5.2 MaxEntGrams

MaxEntGrams[4] is a variant of the Hayes and Wilson learner that offers time and space improvements over the original algorithm by training on an n-gram model of the training data rather than the data itself. For a more detailed comparison of the two models, see Nelson (2022). This model also produces unnormalized log probabilities.

5.3 Smoothed bigram model

This model is included as a baseline. It defines the probability of a word as in Eq. 4, but with conditional probabilities estimated from counts with additive smoothing:

$$p(\sigma_t|\sigma_{t-1}) = \frac{C(\sigma_{t-1}, \sigma_t) + 1}{C(\sigma_{t-1}) + d}, \qquad (7)$$

where $C(\sigma_{t-1}, \sigma_t)$ is the count of the sequence $\sigma_{t-1}\sigma_t$ in the training data, $C(\sigma_{t-1})$ the count of σ_{t-1}, and d the number of distinct segments. This score is reported as a log probability.

This model operates on segmental representations, and thus cannot generalize along featural dimensions. Additive smoothing mitigates this somewhat by assigning every segmental bigram an initial pseudo-count of 1. This ensures that forms containing bigrams that are not in the training data are assigned low, rather than zero, probabilities.

5.4 Summary of models

We do not consider every possible permutation of the models and featurizations above, but present the set shown in Table 3. In particular, we report only a single combination of the models presented in Nelson (2022). In addition to comparing the models themselves, we also focus our analysis on the dimensions of *continuous vs. discrete features* and *phonetic vs. distributional features*.

6 Model comparison on English onset sequences

We compare the performance of the log-bilinear model against the models above on the problem of learning restrictions on onset clusters in English. This problem has been extensively studied in the context of the **Sonority Sequencing Principle** (SSP): the cross-linguistic preference for syllable onsets that monotonically increase in sonority and codas that monotonically decrease in sonority

[4] https://github.com/MaxAndrewNelson/PhoneGraphs

Model	Featurization
Smoothed bigram	N/A
Hayes & Wilson	Discrete phonetic
Hayes & Wilson	Discrete distributional (Mayer)
Bilinear	Continuous distributional (PMI)
Bilinear	Discrete phonetic
Bilinear	Discrete distributional (Mayer)
MaxEntGrams	Discrete distributional (SC COV)

Table 3: Models to be tested

(Selkirk, 1984). Sensitivity to the SSP has been found in many experimental studies, and it has been argued that it constitutes an innate phonological bias (Berent et al., 2008, 2011). Computational studies have shown that phonotactic learning models operating on lexical statistics can learn generalizations about the SSP that align with human behavior, despite having no biases towards SSP-conforming onsets (Daland et al., 2011; Mayer and Nelson, 2020; Nelson, 2022). However, models that incorporate both prior bias and statistical learning have been shown to account better for SSP judgments than either does individually, suggesting a role for both bias and experience (Jarosz and Rysling, 2017; Jarosz, 2017/8). We do not employ this dataset here to make any strong claims about the innateness of the SSP, but rather because it has been used to compare the performance of phonotactic models in previous work.

The training data for all models was the English onset corpus from Hayes and Wilson (2008). This consists of all word-initial onsets from the CMU Pronouncing Dictionary (Weide et al., 1998, `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`): thus each word type in the dictionary contributes a single token to the corpus. Hayes and Wilson sanitize the corpus by removing "exotic" onsets such as [zw], [sf], and [pw] that are unlikely to be encountered by language learners, and by assuming that [j] off-glides are part of the nucleus. We used this dataset to construct the distributional embeddings and to fit the parameters of each model. Following Nelson (2022), the distributional embeddings were calculated over the set of unique onsets (or onset types).

We did a hyperparameter search using cross-validation to determine the learning rate and batch size used to train the log-bilinear model. We considered the values $[32, 64, 128, 256, 512, 1024, 2048, 4096]$ for batch size and $[0.1, 0.01, 0.001, 0.0001]$ for the learning rate. A batch size of 64 and learning rate of 0.001 led to the optimal fit.

We restricted the H&W learner to bigram constraints, allowed it to learn a maximum of 300 constraints, and used the default maximum Observed/Expected threshold of 0.3.

The models were tested on the experimental data from Daland et al. (2011). These data consist of Likert ratings given by 48 participants to a set of 96 nonce words beginning with 48 different onsets. Daland et al. (2011) group the onsets into three different classes: *attested* onsets, which are common in English, *marginal* onsets, which are attested but uncommon, and *unattested* onsets. Following Nelson (2022), we train and test on the onsets in isolation (i.e., the data consist of forms like "sm", "pl", etc.). Each onset is represented by two data points corresponding to two tails the onset was attached to in the Daland et al. study. The onsets are shown in Table 4.

Attested	Marginal	Unattested
tw tr sw	gw ʃl	pw zr mr
ʃr pr pl	vw ʃw	tl dn km
kw kr kl	ʃn ʃm	fn ml nl
gr gl fr	vl bw	dg pk lm
fl dr br	dw fw	ln rl lt
bl sn sm	vr θw	rn rd rg

Table 4: Onsets from Daland et al. (2011).

The trained models assigned scores to the test data according to their onsets. We evaluated model performance by looking at the correlation of scores assigned by each model to the Likert ratings provided by human participants. Following Daland et al. (2011), we look at correlations within the attested/marginal/unattested onset groups, as well as overall correlation. We report both Pearson's r, which captures relative differences in well-formedness but is sensitive to non-linearity between model scores and human judgments, and Kendall's τ, which is not sensitive to non-linearity but only considers the rank ordering of points (see Albright, 2009).

The results are shown in Table 5. The two most successful models are the Hayes & Wilson learner with discrete phonetic features, and the log-bilinear model with continuous distributional features: these have the two highest overall τ correla-

Model	Featurization	Overall		Attested		Marginal		Unattested	
		r	τ	r	τ	r	τ	r	τ
Smoothed bigram	segments	**0.877**	0.669	0.509	0.244	0.274	-0.004	0.470	0.280
MaxEntGrams	discrete dist.	0.753	0.610	0.424	0.282	0.212	0.171	**0.583**	0.417
H&W	discrete phon.	0.740	0.674	0.533	0.261	**0.422**	**0.301**	0.459	0.374
	discrete dist.	0.818	0.634	0.540	0.244	-0.012	-0.049	0.547	0.421
Bilinear	discrete phon.	0.785	0.646	0.446	0.215	0.367	0.247	0.525	0.377
	discrete dist.	0.757	0.572	0.520	0.296	0.021	0.067	0.523	0.309
	continuous dist.	0.699	**0.694**	**0.611**	**0.332**	0.247	0.201	0.562	**0.465**

Table 5: Model comparison using Pearson's r and Kendall's τ to correlate model scores with acceptability ratings for English onsets. The correlation value for the top performing model in each category is bolded.

tions and achieve the highest τ correlations in each of the four categories. Fig. 1 shows the relationship between model scores and human Likert ratings.

The high performance of the bilinear model with continuous distributional features when compared against the same model with discretized distributional features shows that the continuous features contain phonotactically relevant information which is lost under discretization.

It is also interesting to note that the distributional models achieve the highest correlations for all but the marginal forms, which are best captured by models with phonetic features. This may suggest that the relative importance of distributional vs. phonetic information varies in different contexts, but more research will be needed to see if this observation is borne out more generally.[5]

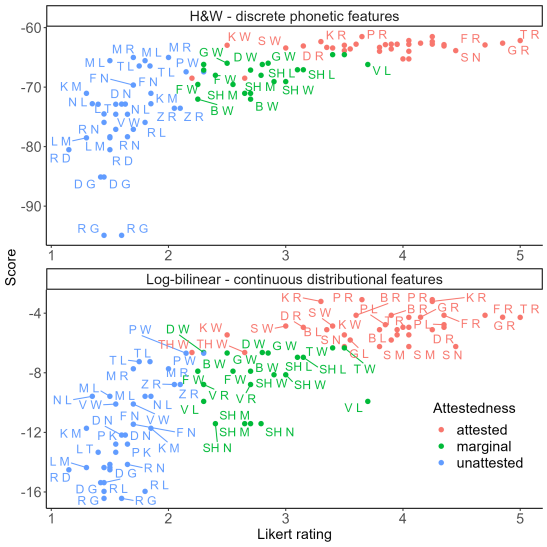


Figure 1: Comparison of the predictions of the two most successful models against human Likert ratings.

7 Conclusion

This paper has presented a log-bilinear model of phonotactics that can incorporate continuous representations of phonological information, bypassing the discretization steps used in previous work. The results of a modeling study showed that this model achieves competitive performance in predicting experimental judgments of English onsets. This model opens up the space of possibilities for phonotactic modeling by removing requirements for discrete representations, allowing greater compatibility with standard distributional learning techniques.

The log-bilinear model is also compatible with continuous representations proposed in other contexts, such as on the basis of phonetic measurements (Mielke, 2012). The model could be used to implement a model of phonotactics that operates directly on these representations, providing insight into the role of fine-grained phonetic detail in phonotactic judgments. More generally, different feature systems may be compared within the log-bilinear framework, and the log-bilinear model itself can be used to generate optimized distributional vector representations of segments: this is the method used to create word2vec vectors when applied to text data (Mikolov et al., 2013; Goldberg and Levy, 2014).

Finally, the log-bilinear model can be straight-

[5]The high performance of the smoothed bigram model on the overall Pearson's correlation is likely due to a strong numerical match with the acceptability ratings of the attested forms, as noted by Daland et al. (2011): performance on unattested and marginal categories, and using Kendall's τ, is substantially worse.

forwardly applied to larger contexts than bigram windows, including autosegmental or tier-based contexts (Goldsmith, 1976; Heinz et al., 2011), by appropriately defining the context representation. The flexibility, relative simplicity, and performance of this model make it a promising framework for studying phonotactic learning and representations.

References

Adam Albright. 2009. Feature-based generalisation as a source of gradient acceptability. *Phonology*, 26(1):9–41.

Diana Archangeli and Douglas Pulleyblank. 2018. Phonology as an emergent system. In S.J. Hannahs and Anna R.K. Bosch, editors, *The Routledge Handbook of Phonological Theory*, pages 476–503. Routledge, London.

Diana Archangeli and Douglas Pulleyblank. 2022. *Emergent phonology*. Language Science Press, Berlin.

Iris Berent, Katherine Harder, and Tracy Lennertz. 2011. Phonological universals in early childhood: Evidence from sonority restrictions. *Language Acquisition*, 18(4):281–293.

Iris Berent, Tracy Lennertz, Jongho Jun, Miguel A. Moreno, and Paul Smolensky. 2008. Language universals in human brains. *Proceedings of the National Academy of Sciences*, 105:5321–5325.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural language processing with Python*. O'Really Media Inc.

Paul Boersma and Joe Pater. 2016. Convergence properties of a gradual learning algorithm for Harmonic Grammar. In John J. McCarthy and Joe Pater, editors, *Harmonic Grammar and Harmonic Serialism*. Equinox, Sheffield.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.

Noam Chomsky and Morris Halle. 1965. Some controversial questions in phonological theory. *Journal of Linguistics*, 1(2):97–138.

Kenneth W. Church and Patrick Hanks. 1990. Word association, norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

John Coleman and Janet Pierrehumbert. 1997. Stochastic phonological grammars and acceptability. In John Coleman, editor, *Proceedings of the 3rd Meeting of the ACL Special Interest Group in Computational Phonology*, pages 49–56. Association for Computational Linguistics, Somerset, NJ.

Robert Daland. 2015. Long words in maximum entropy phonotactic grammars. *Phonology*, 32(3):353–383.

Robert Daland, Bruce Hayes, James White, Marc Garellek, Andreas Davis, and Ingrid Normann. 2011. Explaining sonority projection effects. *Phonology*, 28:197–234.

Stephen A. Della Pietra, Vincent J. Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions: Pattern Analysis and Machine Intelligence*, 19:380–393.

Richard Futrell. 2022. Estimating word co-occurrence probabilities from pretrained static embeddings using a log-bilinear model. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 54–60, Dublin, Ireland. Association for Computational Linguistics.

Richard Futrell, Adam Albright, Peter Graff, and Timothy J. O'Donnell. 2017. A generative model of phonotactics. *Transactions of the Association for Computational Linguistics*, 5:73–86.

Gillian Gallagher. 2019. Phonotactic knowledge and phonetically natural classes. *Phonology*, 36(1):37–60.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

John Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

John Goldsmith and Aris Xanthos. 2009. Learning phonological categories. *Language*, 85(1):4–38.

Maria Gouskova and Gillian Gallagher. 2020. Inducing nonlocal constraints from baseline phonotactics. *Natural Language and Linguistic Theory*, 38(1):77–116.

Bruce Hayes. 2009. *Introductory Phonology*. Wiley-Blackwell, Malden, MA.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Bruce Hayes, Kie Zuraw, Peter Siptar, and Zsuzsa Londe. 2009. Natural and unnatural constraints in Hungarian vowel harmony. *Language*, 85:822–863.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Gaja Jarosz. 2017/8. Defying the stimulus: acquisition of complex onsets in Polish. *Phonology*, 34(2):269–298.

Gaja Jarosz and Amanda Rysling. 2017. Sonority sequencing in Polish: the combined roles of prior bias and experience. In Karen Jesney, Charlie O'Hara, Caitlin Smith, and Rachel Walker, editors, *Supplemental Proceedings of the 2016 Annual Meeting on Phonology*. Linguistic Society of America, Washington, DC.

Daniel Jurafsky and James H. Martin. 2023. Speech and language processing (3rd edition). https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27.

Connor Mayer. 2020. An algorithm for learning phonological classes from distributional similarity. *Phonology*, 37(1):91–131.

Connor Mayer and Robert Daland. 2020. A method for projecting features from observed sets of phonological classes. *Linguistic Inquiry*, 51(4):725–763.

Connor Mayer and Max Nelson. 2020. Phonotactic learning with neural language models. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 291–301, New York, New York. Association for Computational Linguistics.

John J. McCarthy. 1991. Semitic gutturals and distinctive feature theory. *Perspectives on Arabic linguistics*, 3:63–91.

Jeff Mielke. 2008. *The emergence of distinctive features*. Oxford University Press, Oxford.

Jeff Mielke. 2012. A phonetically based metric of sound similarity. *Lingua*, 122(2):145–163.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Nicole Mirea and Klinton Bicknell. 2019. Using LSTMs to assess the obligatoriness of phonological distinctive features for phonotactic learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1595–1605, Florence, Italy. Association for Computational Linguistics.

Andriy Mnih and Geoffrey E Hinton. 2007. Three new graphical models for statistical language modelling. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pages 641–648.

Andriy Mnih and Geoffrey E Hinton. 2008. A scalable hierarchical distributed language model. *Advances in Neural Information Processing Systems*, 21:1081–1088.

Igor Nedjalkov. 1997. *Evenki*. Routledge, London.

Max Nelson. 2022. *Phonotactic learning with distributional representations*. Ph.D. thesis, University of Massachusetts, Amherst.

Carole Paradis and Darlene LaCharité. 2001. Guttural deletion in loanwords. *Phonology*, 18(2):255–300.

Joe Pater. 2009. Weighted constraints in generative linguistics. *Cognitive Science*, 33:999–1035.

Karen Rice and Peter Avery. 1989. On the interaction between sonorancy and voicing. *Toronto Working Papers in Linguistics*, 10.

Elisabeth Selkirk. 1984. On the major class features and syllable theory. In Mark Aronoff and Richard T. Oehrle, editors, *Language sound structure: Studies in phonology presented to Morris Halle by his teacher and students*, pages 107–113. MIT press, Cambridge, MA.

Miikka Silfverberg, Lingshuang Jack Mao, and Mans Hulden. 2018. Sound analogies with phoneme embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*.

Paul Smolensky and Geraldine Legendre. 2006. *The harmonic mind: From neural computation to optimality theoretic grammar*. MIT Press, Cambridge.

Robert Weide et al. 1998. The Carnegie Mellon pronouncing dictionary. *Release 0.6, www.cs.cmu.edu*.

Colin Wilson and Gillian Gallagher. 2018. Accidental gaps and surface-based phonotactic learning: A case study of South Bolivian Quechua. *Linguistic Inquiry*, 49(3):610–623.

Stochastic harmonic grammars do not peak on the mappings singled out by categorical harmonic grammars

Giorgio Magri
CNRS, MIT
magrigrg@gmail.com

Abstract

A candidate surface phonological realization is called a peak of a probabilistic constraint-based phonological grammar provided it achieves the largest probability mass over its candidate set. Obviously, the set of peaks of a maximum entropy grammar is the categorical harmonic grammar corresponding to the same weights. This paper shows that the set of peaks of a stochastic harmonic grammar instead can be different from the categorical harmonic grammar corresponding to any weights. Thus in particular, maximum entropy and stochastic harmonic grammars can peak on different candidates.

Maximum Entropy grammars (ME; Goldwater and Johnson, 2003; Hayes and Wilson, 2008) and Noisy or Stochastic Harmonic Grammars (SHG; Boersma and Pater, 2016) are both probabilistic extensions of categorical Harmonic Grammars (HG; Legendre et al., 1990b,a; Pater, 2009). A growing body of literature tries to pull apart these two probabilistic frameworks. One line of research compares ME and SHG in terms of their ability to fit specific patterns of data given specific choices of candidates and constraints (Zuraw and Hayes, 2017; Smith and Pater, 2020; Breiss and Albright, 2022). Another line of research compares their typological predictions independently of the choice of the constraints, by characterizing the uniform probability inequalities they predict (Anttila and Magri, 2018; Anttila et al., 2019; Magri and Anttila, 2023).

This paper compares ME and SHG in terms of their probability peaks, namely the candidates to which they assign largest probability masses, as formalized in section 1. Obviously, the peaks of the ME grammar corresponding to some non-negative weights are the winners singled out by the categorical HG grammar corresponding to the same weights, no matter what the constraint set looks like, as illustrated in section 2. In other words, ME grammars peak on HG winners. Crucially, this property does not extend from ME to SHG, as discussed in section 3. Indeed, section 4 constructs an example of SHG grammar whose peaks cannot be described as the HG winners corresponding to any non-negative weights. In other words, SHG grammars do not necessarily peak on HG winners. It follows in particular that ME and SHG grammars can peak on different candidates.

The proposed counterexample is somewhat contrived and no simpler counterexamples seem readily available. It is therefore improbable that we would ever "stumble" into one such counterexample by simply "playing" with SHG phonology. This result about SHG peaks thus shows that only mathematical analysis can reveal subtle properties of probabilistic phonological models—which is one of the main goals of linguistic theory.

1 Peaks of probabilistic grammars

A phonological mapping is a pair (x, y) consisting of an underlying form x and a corresponding surface realization y. Gen denotes the set of mappings relevant for the description of the phonological system of interest (Prince and Smolensky, 1993/2004). $Gen(\mathsf{x})$ denotes the set of candidate surface realizations y such that the mapping (x, y) belongs to Gen. We allow Gen to list countably infinitely many underlying forms. But we require a candidate set $Gen(\mathsf{x})$ to be finite to avoid the technicalities needed to define probability mass functions on infinite sets.

A **categorical grammar** G assigns to an underlying form x a unique "winner" surface realization y from the candidate set $Gen(\mathsf{x})$. Thus, we require categorical grammars to be **strict**: they specify a unique surface realization per underlying form. On the other hand, we allow categorical grammars to be **partial**: they might fail to specify any surface realization for a given underlying form. HG grammars recalled below are indeed usually defined as strict and partial.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 269-277.
Amherst, Massachusetts, June 15-17, 2023

A **probabilistic grammar** G assigns to each mapping (x, y) listed by Gen a number $G(\mathsf{y} \,|\, \mathsf{x})$ that is interpreted as the probability that the underlying form x is realized as the surface candidate y. This probabilistic interpretation requires these numbers $G(\mathsf{y} \,|\, \mathsf{x})$ to be non-negative and normalized across the candidate set $Gen(\mathsf{x})$ of each underlying form x, namely $\sum_{\mathsf{y} \in Gen(\mathsf{x})} G(\mathsf{y} \,|\, \mathsf{x}) = 1$.

We say that a mapping (x, y) is a **peak** of a probabilistic grammar G provided y is assigned a larger probability mass than any other candidate z of the underlying form x, as stated in (1).

$$G(\mathsf{y} \,|\, \mathsf{x}) > \max_{\substack{\mathsf{z} \in Gen(\mathsf{x}) \\ \mathsf{z} \neq \mathsf{y}}} G(\mathsf{z} \,|\, \mathsf{x}) \tag{1}$$

The set of candidates with peak probabilities can be interpreted as a categorical grammar. This categorical grammar is strict, because condition (1) features a strict inequality, whereby at most one candidate per underlying form qualifies as a peak. Furthermore, this categorical grammar is partial, because multiple candidates can tie for the largest probability, whereby none qualifies as a peak.

Intuitively, these candidates that are assigned the largest probability masses are those that are deemed most important by a probabilistic grammar. The set of these most important candidates with peak probabilities thus ought to capture some important information about the probabilistic grammar. As a first stub at analyzing a complex probabilistic grammar, it thus makes sense to analyze the corresponding categorical grammar of peaks.

2 ME peaks are HG winners

To illustrate the definitions in the preceding section, we consider a set $\boldsymbol{C}$ consisting of a finite number n of constraints C_k. We denote by $C_k(\mathsf{x}, \mathsf{y})$ the number of violations assigned by constraint C_k to a mapping (x, y) from Gen. We assign to each constraint C_k a non-negative weight w_k. A candidate y is the winner surface realization of an underlying form x provided it satisfies condition (2). It says that the candidate y violates the constraints less than any other candidate z because the weighted sum of the constraint violations of y is strictly smaller. The categorical grammar G that singles out such winner candidates is the **HG** grammar corresponding to the weight vector $\boldsymbol{w} = (w_1, \ldots, w_n)$. It is strict, because (2) features a strict inequality. It can be partial, in case two or more candidates tie for the smallest weighted sum of constraint violations.

$$\sum_{k=1}^{n} w_k C_k(\mathsf{x}, \mathsf{y}) < \min_{\substack{\mathsf{z} \in Gen(\mathsf{x}) \\ \mathsf{z} \neq \mathsf{y}}} \sum_{k=1}^{n} w_k C_k(\mathsf{x}, \mathsf{z}) \tag{2}$$

We can also use the constraint set $\boldsymbol{C}$ and the weight vector $\boldsymbol{w}$ to define a probabilistic grammar though condition (3). It says that the probability $G(\mathsf{y} \,|\, \mathsf{x})$ that an underlying form x is realized as a candidate y is the exponential of the opposite of the weighted sum of constraint violations of the mapping (x, y), divided by a quantity $Z(\mathsf{x})$ that ensures normalization over the candidate set $Gen(\mathsf{x})$. The resulting probabilistic grammar G is the **ME** grammar corresponding to the weight vector $\boldsymbol{w}$.

$$G(\mathsf{y} \,|\, \mathsf{x}) = \frac{1}{Z(\mathsf{x})} \exp\left\{ -\sum_{k=1}^{n} w_k C_k(\mathsf{x}, \mathsf{y}) \right\} \tag{3}$$

The normalization constant $Z(\mathsf{x})$ depends on the underlying form x but not on the candidate y. Furthermore, the definition (1) of probability peaks only compares probabilities within the same candidate set. It follows that a mapping (x, y) qualifies as a peak of the ME grammar (3) corresponding to the weight vector $\boldsymbol{w}$ if and only if (x, y) belongs to the HG grammar (2) corresponding to the same weight vector $\boldsymbol{w}$. In other words, HG grammars single out the peaks of ME grammars.

3 SHG peaks are not HG winners

Let p_w be a uni-dimensional probability density function that starts at a point w, in the sense that it is equal to zero at the left of w. Here are some natural examples of such a density ($\mathbb{I}_S$ is the indicator function of the set S):

- the uniform density $p_w^{\text{unif}}(v) = \mathbb{I}_{[w,w+1]}(v)$;
- the exponential density $p_w^{\text{exp}}(v) = \exp(w - v)\mathbb{I}_{[w,+\infty]}(v)$;
- the half-gaussian density $p_w^{\text{gauss}}(v) = \frac{2\exp\left[-(v-w)^2/2\right]}{\sqrt{2\pi}}\mathbb{I}_{[w,+\infty]}(v)$

Given a constraint set $\boldsymbol{C}$ and a non-negative weight vector $\boldsymbol{w} = (w_1, \ldots, w_n)$, the corresponding **SHG** grammar assigns to a mapping (x, y) the probability of sampling a weight vector $\boldsymbol{v}$ according to $\mathbf{p}_{\boldsymbol{w}} = p_{w_1} \cdot \ldots \cdot p_{w_n}$ such that y qualifies as an HG winner

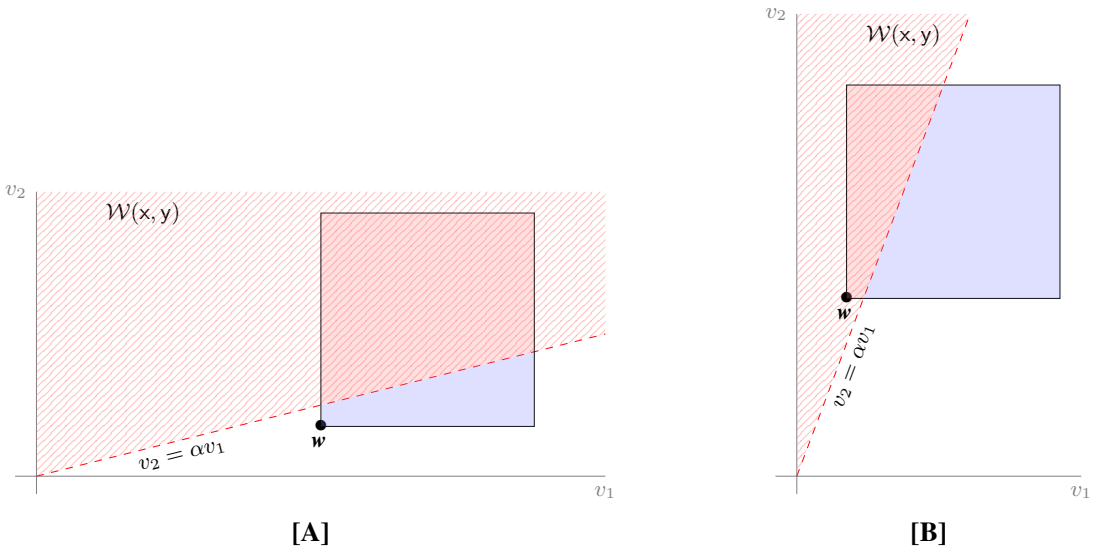


Figure 1

corresponding to this weight vector $\boldsymbol{v}$ in the sense of condition (2) above. The assumption that the weights w_k are non-negative and that p_{w_k} starts at w_k ensures that the probability of sampling vectors $\boldsymbol{v}$ with negative components is zero.[1]

To understand intuitively why SHG peaks are not necessarily HG winners, let us consider the following simplest case. *Gen* lists only two candidate surface realizations y and z for an underlying form x. The constraint set consists of only $n = 2$ constraints. The set $\mathcal{W}(\mathsf{x},\mathsf{y})$ of weight vectors $\boldsymbol{v} = (v_1, v_2)$ such that the weighted sum of constraint violations of y is smaller than that of z is the dashed red cone in figure 1 described by the inequality $v_2 > \alpha v_1$, for some $\alpha > 0$. The SHG grammar corresponding to a weight vector $\boldsymbol{w}$ is implemented with the uniform density that concentrates the probability mass on the square that starts at the weight vector $\boldsymbol{w}$. This square is split into two halves by the red dashed cone $\mathcal{W}(\mathsf{x},\mathsf{y})$. The area of the solid red half of the square that sits within the cone $\mathcal{W}(\mathsf{x},\mathsf{y})$ is the probability mass assigned by our SHG grammar to the mapping (x,y). The area of the remaining solid blue half that sits outside of $\mathcal{W}(\mathsf{x},\mathsf{y})$ is the probability mass assigned to (x,z).

The weight vector $\boldsymbol{w}$ in figure 1A sits outside of the cone $\mathcal{W}(\mathsf{x},\mathsf{y})$. Thus, the HG grammar corresponding to $\boldsymbol{w}$ does not contain the mapping (x,y). Yet, $\boldsymbol{w}$ sits so close to the border of the cone $\mathcal{W}(\mathsf{x},\mathsf{y})$ that the red solid area is larger than the blue solid area. Thus, our mapping (x,y) is a peak of the SHG grammar corresponding to the weight vector $\boldsymbol{w}$, because (x,y) receives a larger probability mass than (x,z). In conclusion, a peak of an SHG grammar might not belong to the HG grammar corresponding to the same weight vector.

Figure 1B illustrates the reverse scenario. The HG grammar contains the mapping (x,y) because the weight vector $\boldsymbol{w}$ sits inside the cone $\mathcal{W}(\mathsf{x},\mathsf{y})$. Yet, $\boldsymbol{w}$ sits so close to the border of the cone that the mapping (x,y) does not count as a peak of the SHG grammar because the red solid area is smaller than the blue solid area. In conclusion, a mapping of an HG grammar might not be a peak of the SHG grammar corresponding to the same weight vector.

These mismatches between SHG peaks and HG mappings are only possible when the border of the cone $\mathcal{W}(\mathsf{x},\mathsf{y})$ is **less** tilted than the diagonal because $\alpha < 1$ as in figure 1A; or it is **more** titled than the diagonal because $\alpha > 1$ as in figure 1B. These mismatches are not possible when instead the border of the cone $\mathcal{W}(\mathsf{x},\mathsf{y})$ coincides with the

[1] The implementation of probabilistic constraint-based phonology that I call here "stochastic" HG is slightly different from what Boersma and Pater (2016) call "noisy" HG, because the two implementations differ for the strategy they adopt to avoid sampling zero weights. In SHG, zero weights are avoided by sampling according to a density p_w that starts at a positive value w. In NHG, zero weights are avoided by clipping at zero or by re-sampling (Hayes and Kaplan 2023). Furthermore, the term "stochastic" HG makes it explicit that the resulting framework is a probabilistic extension of categorical HG in exactly the same way that Stochastic OT is a probabilistic extension of categorical OT (Boersma 1997, 1998). Finally, the term "noisy" is traditionally used to qualify the training data, while "stochastic" is used to single out algorithms (and thus grammars) that are non-deterministic.

diagonal because $\alpha = 1$. In this case, the red solid area is larger (smaller) than the blue solid area if and only if the weight vector $\boldsymbol{w}$ sits inside (outside) of the cone $\mathcal{W}(\mathsf{x}, \mathsf{y})$, no matter how close $\boldsymbol{w}$ is to the diagonal border of the cone. As a result, (x, y) is an SHG peak if and only if it belongs to the HG grammar corresponding to the same weights. We will use this observation in subsection 4.1 below.

The considerations developed so far for the uniform density based on elementary geometric considerations extend to other densities. To illustrate, let us consider the exponential density. We start with the case where the border of the cone $\mathcal{W}(\mathsf{x}, \mathsf{y})$ is **less tilted** than the diagonal as in figure 1A, say because $\alpha = 1/2$. We focus on weight vectors $\boldsymbol{w} = (w_1, w_2)$ that sit outside of this cone because they have a negative "distance" $\xi = w_2 - \alpha w_1 < 0$ from the border of the cone. The SHG probability mass of our mapping (x, y) is easily computed in closed form by integrating the exponential function. Figure 2A plots this SHG probability mass (on the vertical axis) as a function of the "distance" $\xi < 0$ (on the horizontal axis). When ξ is between $\log(3/4) \simeq -0.2877$ and zero, the weight vector $\boldsymbol{w}$ sits outside of the cone $\mathcal{W}(\mathsf{x}, \mathsf{y})$, whereby the mapping (x, y) does not belong to the corresponding HG grammar. Yet (x, y) is a peak of the corresponding SHG grammar, because the SHG probability mass of y is larger than 0.5, and therefore larger than the SHG probability mass of z.

Analogously, let us consider the case where the border of the cone $\mathcal{W}(\mathsf{x}, \mathsf{y})$ is **more tilted** than the diagonal as in figure 1B, say because $\alpha = 2$. We focus on weight vectors $\boldsymbol{w} = (w_1, w_2)$ that sit inside this cone because they have a positive "distance" $\xi = w_2 - \alpha w_1 > 0$ from the border of the cone. Figure 2B plots the SHG probability mass of our mapping (x, y) as a function of the "distance" $\xi > 0$. When ξ is between zero and $\log(16/9) \simeq 0.5754$, the weight vector $\boldsymbol{w}$ sits inside the cone $\mathcal{W}(\mathsf{x}, \mathsf{y})$, whereby the mapping (x, y) does belong to the corresponding HG grammar. Yet (x, y) is not a peak of the corresponding SHG grammar, because the SHG probability mass of y is smaller than 0.5, and therefore smaller than the SHG probability mass of z.

These considerations show that the mappings singled out by the HG grammar corresponding to some weight vector $\boldsymbol{w}$ are not necessarily the peaks of the SHG grammar corresponding to the same weight vector $\boldsymbol{w}$. Yet, it can be shown (through a different line of analysis that falls outside of the scope of this paper), that the mappings singled out by the HG grammar corresponding to some weight vector $\boldsymbol{w}$ are always the peaks of the SHG grammar corresponding to a possibly different weight vector $\boldsymbol{w}'$. What about the reverse? Despite the mismatches between SHG peaks and HG mappings documented above, is it the case that the peaks of the SHG grammar corresponding to some weight vector $\boldsymbol{w}$ are always the mappings singled out by the HG grammar corresponding to a possibly different weight vector $\boldsymbol{w}'$? The next section provides a negative answer to this question by constructing an SHG grammar whose set of peaks is not an HG grammar, no matter the choice of the weights.

4 Counterexample

To construct the simplest possible counterexample, we assume that Gen lists only three underlying forms x_1, x_2, and x_3 and endows each of them with only two candidates y_i and z_i, as in (4)

$$Gen = \left\{ \begin{array}{ccc} (\mathsf{x}_1, \mathsf{y}_1) & (\mathsf{x}_2, \mathsf{y}_2) & (\mathsf{x}_3, \mathsf{y}_3) \\ (\mathsf{x}_1, \mathsf{z}_1) & (\mathsf{x}_2, \mathsf{z}_2) & (\mathsf{x}_3, \mathsf{z}_3) \end{array} \right\} \quad (4)$$

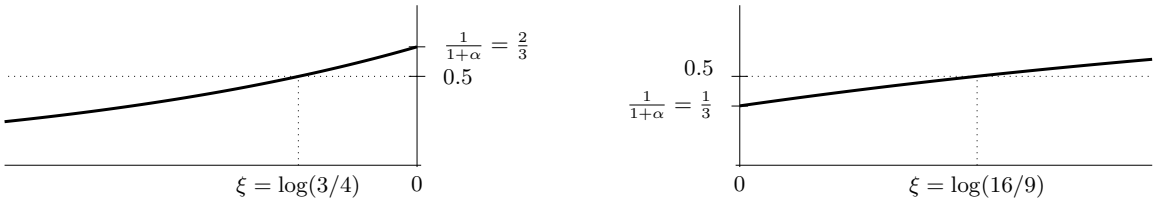
The constraint set $\boldsymbol{C}$ consists of $n = 3$ constraints C_1, C_2, and C_3 that yield the violation profiles in (5). Actual numbers of constraint violations do not matter. What does matter for the counterexample are the ratios of the differences between the numbers of violations of two candidates, as shown in appendix E. To illustrate, it does not matter that C_1 and C_3 assign 33 and 0 violations to y_3 and 0 and 200 violations to z_3. What does matter is that the ratio between the differences $C_1(\mathsf{x}_3, \mathsf{y}_3) - C_1(\mathsf{x}_3, \mathsf{z}_3)$ and $C_3(\mathsf{x}_3, \mathsf{z}_3) - C_3(\mathsf{x}_3, \mathsf{y}_3)$ is equal to $33/200 = 0.165$. These large numbers 33 and 200 are needed to express a small value 0.165 as the ratio 33/200 between two integers.

	C_1	C_2	C_3
$(\mathsf{x}_1, \mathsf{y}_1)$	0	5	0
$(\mathsf{x}_1, \mathsf{z}_1)$	2	0	0
$(\mathsf{x}_2, \mathsf{y}_2)$	0	0	5
$(\mathsf{x}_2, \mathsf{z}_2)$	0	2	0
$(\mathsf{x}_3, \mathsf{y}_3)$	33	0	0
$(\mathsf{x}_3, \mathsf{z}_3)$	0	0	200

(5)

Finally, the vector $\boldsymbol{w} = (w_1, w_2, w_3)$ of non-negative weights is chosen carefully as in (6).

$$\begin{array}{rcl} w_1 & = & 4.21734890439 \\ w_2 & = & 1.3195643695 \\ w_3 & = & 0.160450555542 \end{array} \quad (6)$$



[A] Case $\alpha = 1/2$ **[B]** Case $\alpha = 2$

Figure 2

We implement the SHG grammar with the exponential density. When this SHG grammar is called for 100,000 times on each of the three underlying forms x_1, x_2, and x_3, it returns the surface forms y_1, y_2, and y_3 with the frequencies in (7).

$$\begin{aligned} G_{\boldsymbol{w}}^{\text{SHG}}(\mathsf{y}_1 \,|\, \mathsf{x}_1) &= 0.50566 \\ G_{\boldsymbol{w}}^{\text{SHG}}(\mathsf{y}_2 \,|\, \mathsf{x}_2) &= 0.50637 \\ G_{\boldsymbol{w}}^{\text{SHG}}(\mathsf{y}_3 \,|\, \mathsf{x}_3) &= 0.50126 \end{aligned} \tag{7}$$

Since these frequencies are (close to but strictly) larger than 0.5, the categorical grammar of peaks of the SHG grammar considered is the grammar G in (8). Crucially, we will see that this grammar G is not an HG grammar corresponding to any choice of non-negative constraint weights.

$$G = \big\{(\mathsf{x}_1, \mathsf{y}_1), (\mathsf{x}_2, \mathsf{y}_2), (\mathsf{x}_3, \mathsf{y}_3)\big\} \tag{8}$$

In conclusion, we have constructed an SHG grammars whose set of peaks cannot be construed as any HG grammar. The rest of this section explains in detail how the counterexample has been constructed, by motivating the choice of the violation profiles in (5) and of the weights in (6).

4.1 First step

We need to define the $n = 3$ constraints in such a way that the grammar G in (8) is not an HG grammar. As above, let us denote by $\mathcal{W}(\mathsf{x}_i, \mathsf{y}_i)$ the cone of those non-negative weight vectors $\boldsymbol{v} = (v_1, v_2, v_3)$ that declare y_i the winner surface realization of the underlying form x_i. A simple strategy to achieve our goal is to define the constraints so that these three cones are as in (9). In fact, the grammar G in (8) qualifies as an HG grammar only if some non-negative weight vector $\boldsymbol{v} = (v_1, v_2, v_3)$ belongs simultaneously to all three cones. And that is impossible. Because a weight vector that belongs to both cones $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$ and $\mathcal{W}(\mathsf{x}_2, \mathsf{y}_2)$ must satisfy both inequalities $v_1 > v_2$ and $v_2 > v_3$. By transitivity, it must also satisfy the inequality $v_1 > v_3$. Hence, our weight vector cannot belong to the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$.

$$\begin{aligned} \mathcal{W}(\mathsf{x}_1, \mathsf{y}_1) &= \big\{\boldsymbol{v} \,\big|\, v_1 > v_2\big\} \\ \mathcal{W}(\mathsf{x}_2, \mathsf{y}_2) &= \big\{\boldsymbol{v} \,\big|\, v_2 > v_3\big\} \\ \mathcal{W}(\mathsf{x}_3, \mathsf{y}_3) &= \big\{\boldsymbol{v} \,\big|\, v_1 < v_3\big\} \end{aligned} \tag{9}$$

Unfortunately, the borders of the cones in (9) are all diagonal. As discussed in section 3 above, we get no mismatches between SHG peaks and HG mappings in this case. Thus, I make the borders non-diagonal by replacing the cones in (9) with those in (10), where the steepness of the borders is controlled by the positive coefficients a and α. I use the same coefficient a for both cones $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$ and $\mathcal{W}(\mathsf{x}_2, \mathsf{y}_2)$, as this choice simplifies the analysis without compromising the counterexample.

$$\begin{aligned} \mathcal{W}(\mathsf{x}_1, \mathsf{y}_1) &= \big\{\boldsymbol{v} \,\big|\, v_1 > a\, v_2\big\} \\ \mathcal{W}(\mathsf{x}_2, \mathsf{y}_2) &= \big\{\boldsymbol{v} \,\big|\, v_2 > a\, v_3\big\} \\ \mathcal{W}(\mathsf{x}_3, \mathsf{y}_3) &= \big\{\boldsymbol{v} \,\big|\, v_3 > \alpha\, v_1\big\} \end{aligned} \tag{10}$$

As for the HG-hood of the grammar G in (8), the replacement of our initial guess (9) with the refined guess (10) changes nothing because of the following lemma, verified in appendix A.

Lemma 1 *Suppose that the positive coefficients $a, \alpha > 0$ satisfy condition (11).*

$$a^2 \alpha \geq 1 \tag{11}$$

No weight vector $\boldsymbol{v} = (v_1, v_2, v_3)$ belongs simultaneously to the three cones in (10), whereby the grammar G in (8) is not an HG grammar.

4.2 Second step

We now want to construct a non-negative weight vector $\boldsymbol{w} = (w_1, w_2, w_3)$ such that the peaks of the corresponding SHG grammar are indeed the three mappings singled out by the grammar G in (8). As discussed in the preceding subsection, this weight vector $\boldsymbol{w}$ cannot belong simultaneously to all three cones in (10). For concreteness, we assume that the

weight vector $\boldsymbol{w} = (w_1, w_2, w_3)$ does not belong to the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ while it does belong to the other two cones $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$ and $\mathcal{W}(\mathsf{x}_2, \mathsf{y}_2)$.

The assumption that the weight vector $\boldsymbol{w}$ sits outside of the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ despite the mapping $(\mathsf{x}_3, \mathsf{y}_3)$ being a peak of the corresponding SHG grammar has two consequences. The first consequence is that the border of the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ must be less tilted than the diagonal, as in figure 1A. In other words, the coefficient α that controls its tiltedness must be small in the sense that $\alpha < 1$. The second consequence is that, although the weight vector $\boldsymbol{w}$ sits outside of the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$, it cannot be too far away from it. Equivalently, although w_3 is smaller than αw_1 (so that $\boldsymbol{w}$ sits outside of $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$), it cannot be too much smaller (so that $\boldsymbol{w}$ sits close to $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$). Not much smaller in the sense that the weights w_1 and w_3 satisfy the inequality $w_3 > \alpha w_1 - A$ for some carefully chosen positive constant $A > 0$. The following lemma says that we need to choose this constant A equal to $\log \frac{2}{1+\alpha}$, as verified in appendix B. Since $\alpha < 1$, this position $A = \log \frac{2}{1+\alpha}$ is positive as desired.

Lemma 2 *Consider a weight vector* $\boldsymbol{w} = (w_1, w_2, w_3)$ *that does not belong to the cone* $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ *because* $w_3 < \alpha w_1$. *The mapping* $(\mathsf{x}_3, \mathsf{y}_3)$ *is a peak of the SHG grammar corresponding to this weight vector* $\boldsymbol{w}$ *provided* $\boldsymbol{w}$ *satisfies (12).*

$$w_3 > \alpha w_1 - \underbrace{\log \frac{2}{1+\alpha}}_{A} \tag{12}$$

Condition (11) together with the assumption $\alpha < 1$ made above entails that the coefficient a that controls the tiltedness of the border of the cone $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$ is large in the sense that $a > 1$. Equivalently, the border of the cone $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$ is steeper than the diagonal. As a result, the assumption that the weight vector $\boldsymbol{w}$ sits inside the cone $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$ by itself does not suffice to ensure that $(\mathsf{x}_1, \mathsf{y}_1)$ is a peak, as shown in figure 1B. We need to make sure that the weight vector $\boldsymbol{w}$ sits well inside this cone $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$, far away from the border. Equivalently, w_1 is not just larger than aw_2 (so that $\boldsymbol{w}$ sits inside $\mathcal{W}(\mathsf{x}_1\mathsf{y}_1)$) but actually quite larger (so that $\boldsymbol{w}$ sits well inside $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$). Quite larger in the sense that the weights w_1 and w_2 satisfy the inequality $w_1 > aw_2 + B$ for some carefully chosen positive constant $B > 0$. The following lemma says that we need to choose this constant B equal to $a \log \frac{2a}{1+a}$, as verified in appendix C. Since $a > 1$, this position $B = a \log \frac{2a}{1+a}$ is positive as desired.

Lemma 3 *Consider a weight vector* $\boldsymbol{w} = (w_1, w_2, w_3)$ *that does belong to the cone* $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$ *because* $w_1 > aw_2$. *The mapping* $(\mathsf{x}_1, \mathsf{y}_1)$ *is a peak of the SHG grammar corresponding to this weight vector* $\boldsymbol{w}$ *provided* $\boldsymbol{w}$ *satisfies (13).*

$$w_1 > aw_2 + \underbrace{a \log \frac{2a}{1+a}}_{B} \tag{13}$$

A completely analogous reasoning shows that condition (14) ensures that the mapping $(\mathsf{x}_2, \mathsf{y}_2)$ is a peak of the SHG grammar corresponding to the weight vector $\boldsymbol{w} = (w_1, w_2, w_3)$.

$$w_2 > aw_3 + a \log \frac{2a}{1+a} \tag{14}$$

4.3 Third step

Do the three inequalities (12), (13), and (14) just obtained admit non-negative solutions $w_1, w_2, w_3 \geq 0$? To answer this question, we use of the following straightforward fact, verified in appendix D.

Lemma 4 *Suppose that* $a^2\alpha > 1$, *as in (11). The following three strict inequalities*

$$\begin{array}{rcl} w_3 & > & \alpha w_1 - A \\ w_1 & > & aw_2 + B \\ w_2 & > & aw_3 + B \end{array} \tag{15}$$

admit non-negative solutions $w_1, w_2, w_3 \geq 0$ *when their coefficients* $a, \alpha > 0$ *and* $A, B \geq 0$ *satisfy the following condition (16).*

$$1 < \frac{A}{\alpha(1+a)B} \tag{16}$$

Indeed, the inequalities (12), (13), and (14) have the shape in (15) with the positions (17).

$$A = \log \frac{2}{1+\alpha}, \qquad B = a \log \frac{2a}{1+a} \tag{17}$$

Condition (16) that ensures that the three inequalities (15) admit non-negative solutions boils down to condition (18) with these positions (17). We conclude that the inequalities (12), (13), (14) admit non-negative solutions $w_1, w_2, w_3 \geq 0$ when the coefficients a, α satisfy condition (18).

$$\frac{1}{\alpha} \log \frac{2}{1+\alpha} - a(1+a) \log \frac{2a}{1+a} > 0 \tag{18}$$

4.4 Fourth step

In conclusion, in order for our counterexample to work, we need to find coefficients $a > 1$ and $0 < \alpha < 1$ that satisfy both conditions (11) and (18). To this end, figure 3 plots in red (blue) the pairs of values (a, α) that satisfy (do not satisfy) condition (18). Furthermore, the black line in figure 3 describes the equation $\alpha = 1/a^2$. The pairs of values (a, α) that satisfy condition (11) thus sit above and at the right of this black line. This figure thus says that a pair of values (a, α) satisfies both conditions (11) and (18) as desired provided it belongs to the narrow band between the black line and the boundary between the red and blue regions. The pair (a, α) in (19) belongs indeed to this narrow band and thus satisfies both conditions (11) and (18).

$$a = 2.5, \qquad \alpha = 0.165 \tag{19}$$

When the constraint violation vectors are defined as in (5), the cones $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$, $\mathcal{W}(\mathsf{x}_2, \mathsf{y}_2)$, and $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ are precisely the cones described by the inequalities in (10) with the coefficients a and α as in (19), because $5/2 = 2.5 = a$ and $33/200 = 0.165 = \alpha$, as verified in appendix E. Finally, the three inequalities (12), (13), and (14) corresponding to the coefficients a and α in (19) admit non-negative solutions w_1, w_2, w_3 such as those in (6), as shown in appendix F, completing the explanation of the counterexample.

5 Conclusions

Categorical grammars can usually be analyzed by exhaustive enumeration and direct inspection of the mappings they contain. Probabilistic grammars instead require more sophisticated analytical tools. A natural idea is to analyze some of the linguistic information captured by a complex probabilistic grammars by analyzing it peaks, namely the candidates that are deemed most important by that probabilistic grammar because assigned the largest probability mass. For a ME grammar, this is easily done: its peaks are the HG winners corresponding to the same weight vector. This paper has shown that the situation is different in SHG: although any HG grammar can be construed as the set of peaks of some SHG grammar, the set of peaks of some SHG grammars cannot be construed as an HG grammar, no matter the choice of the weights. It follows that ME and SHG grammars corresponding to the same weights can peak on different candidates.

Appendix

A Proof of lemma 1

A weight vector $\boldsymbol{v} = (v_1, v_2, v_3)$ that belongs to both cones $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$ and $\mathcal{W}(\mathsf{x}_2, \mathsf{y}_2)$ satisfies both inequalities $v_1 > av_2$ and $v_2 > av_3$. Thus in particular, $\boldsymbol{v}$ satisfies the inequality $v_1 > a^2 v_3$. On the other hand, a weight vector $\boldsymbol{v}$ that belongs to the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ satisfies the inequality $v_1 < v_3/\alpha$. These two inequalities yield $a^2 v_3 < v_3/\alpha$. Since this inequality is strict, v_3 must be strictly positive and can therefore be simplified, yielding $a^2 < \frac{1}{\alpha}$. This conclusion contradicts the assumption (11).

B Proof of lemma 2

We start by establishing the chain of identities in (20). Step (20a) below holds because of the definition of the exponential density. Step (20b) holds because $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ is the cone consisting of the non-negative vectors $\boldsymbol{v} = (v_1, v_2, v_3)$ such that $v_3 \geq \alpha v_1$. Step (20c) holds because of the hypothesis $w_3 \leq \alpha w_1$ that $\boldsymbol{w} = (w_1, w_2, w_3)$ sits outside of the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$. Thus, $v_1 \geq w_1$ entails $\alpha v_1 \geq \alpha w_1 \geq w_3$, whereby $\max\{w_3, \alpha v_1\} = \alpha v_1$. The remaining steps only use the identity $\int e^{-\lambda x} \mathrm{d}x = -\frac{1}{a} e^{-\lambda x}$.

$$\begin{aligned}
&\int_{\mathcal{W}(\mathsf{x}_3,\mathsf{y}_3)} p^{\exp}_{w_1}(v_1)\, p^{\exp}_{w_3}(v_3)\, \mathrm{d}v_1\, \mathrm{d}v_3 = \\
&\overset{(a)}{=} e^{w_1+w_3} \int_{v_1 \geq w_1, v_3 \geq w_3} e^{-v_1-v_3} \mathbb{I}_{\mathcal{W}(\mathsf{x},\mathsf{y})}(v_1, v_3) \mathrm{d}v_1 \mathrm{d}v_3 \\
&\overset{(b)}{=} e^{w_1} e^{w_3} \int_{v_1 \geq w_1} e^{-v_1} \int_{v_3 \geq \max\{w_3, \alpha v_1\}} e^{-v_3} \mathrm{d}v_1 \mathrm{d}v_3 \\
&\overset{(c)}{=} e^{w_1+w_3} \int_{v_1 \geq w_1} e^{-v_1} \int_{v_3 \geq \alpha v_1} e^{-v_3} \mathrm{d}v_3 \mathrm{d}v_1 \\
&= e^{w_1+w_3} \int_{v_1 \geq w_1} e^{-v_1} \left| -e^{-v_3} \right|_{\alpha v_1}^{\infty} \mathrm{d}v_1 \\
&= e^{w_1+w_3} \int_{v_1 \geq w_1} e^{-(1+\alpha)v_1} \mathrm{d}v_1 \\
&= e^{w_1+w_3} \left| -\frac{1}{(1+\alpha)} e^{-(1+\alpha)v_1} \right|_{w_1}^{\infty} \\
&= e^{w_1+w_3} \frac{1}{1+\alpha} e^{-(1+\alpha)w_1} \\
&= \frac{1}{1+\alpha} e^{-\alpha w_1 + w_3}
\end{aligned} \tag{20}$$

The proof of lemma 2 now consists of the chain of equivalences in (21). Step (21a) holds because the underlying form x_3 has only two candidates y_3 and z_3, whereby the probability mass of z_3 is equal to 1 minus the probability mass of y_3. Step

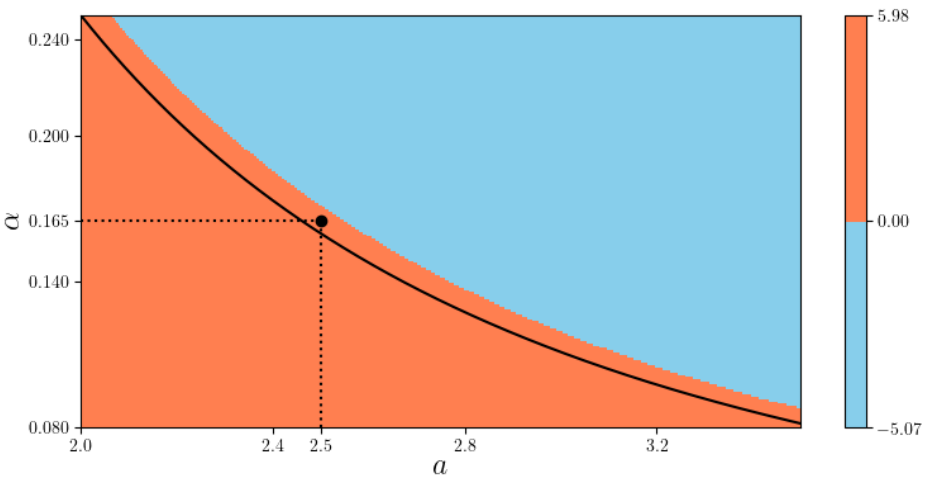


Figure 3

(21b) holds because the probability mass of the mapping $(\mathsf{y}_3 \,|\, \mathsf{x}_3)$ according to the SHG grammar corresponding to the weight vector $\boldsymbol{w}$ is the volume of the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ relative to the product of three exponential densities that start at the weights w_1, w_2, and w_3. Step (21c) holds because the definition (10) of the cone $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$ only looks at the first and third components. Step (21d) holds because of the computation in (20) above.

$$
\begin{aligned}
& G^{\text{SHG}}_{\boldsymbol{w}}(\mathsf{y}_3 \,|\, \mathsf{x}_3) > G^{\text{SHG}}_{\boldsymbol{w}}(\mathsf{z}_3 \,|\, \mathsf{x}_3) \iff \\
& \quad \overset{(a)}{\iff} \quad G^{\text{SHG}}_{\boldsymbol{w}}(\mathsf{y}_3 \,|\, \mathsf{x}_3) > 1 - G^{\text{SHG}}_{\boldsymbol{w}}(\mathsf{y}_3 \,|\, \mathsf{x}_3) \\
& \quad \iff \quad 2 G^{\text{SHG}}_{\boldsymbol{w}}(\mathsf{y}_3 \,|\, \mathsf{x}_3) > 1 \\
& \quad \overset{(b)}{\iff} \quad 2 \int_{\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)} p^{\text{exp}}_{\boldsymbol{w}}(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{v} > 1 \\
& \quad \overset{(c)}{\iff} \quad 2 \int_{\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)} p^{\text{exp}}_{w_1}(v_1) \, p^{\text{exp}}_{w_3}(v_3) \, \mathrm{d}v_1 \, \mathrm{d}v_3 > 1 \\
& \quad \overset{(d)}{\iff} \quad 2 \frac{1}{1+\alpha} \exp(w_3 - \alpha w_1) > 1 \\
& \quad \iff \quad w_3 > \alpha w_1 + \log \frac{1+\alpha}{2}
\end{aligned}
\tag{21}
$$

C Proof of lemma 3

Step (22a) holds as steps (21a-c) above. Step (21b) can be established by reasoning as in (20).

$$
\begin{aligned}
& G^{\text{SHG}}_{\boldsymbol{w}}(\mathsf{y}_1 \,|\, \mathsf{x}_1) > G^{\text{SHG}}_{\boldsymbol{w}}(\mathsf{z}_1 \,|\, \mathsf{x}_1) \\
& \quad \overset{(a)}{\iff} \quad 2 \int_{\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)} p^{\text{exp}}_{w_1}(v_1) \, p^{\text{exp}}_{w_2}(v_2) \, \mathrm{d}v_1 \, \mathrm{d}v_2 \\
& \quad \overset{(b)}{\iff} \quad 2 \left(1 - \frac{a}{1+a} \exp - \frac{w_1 - a w_2}{a} \right) > 1 \\
& \quad \iff \quad w_1 > a w_2 + a \log \frac{2a}{1+a}
\end{aligned}
\tag{22}
$$

D Proof of lemma 4

The positions $w_1 = a w_2 + \epsilon B$ and $w_2 = a w_3 + \epsilon B$ satisfy the second and third inequalities in (15) as long as $\epsilon > 1$. Plugging the latter into the former yields $w_1 = a^2 w_3 + \epsilon B(a+1)$. Plugging the latter into the first inequality in (15) yields (23).

$$
\big(\alpha a^2 - 1\big) w_3 < A - \alpha \epsilon B (1 + a) \tag{23}
$$

The assumption $a^2 \alpha > 1$ means that the coefficient of w_3 on the left-hand side of (23) is strictly positive. Hence, (23) admits a non-negative solution $w_3 \geq 0$ provided $A - \alpha\epsilon(a+1)B > 0$. Equivalently, provided ϵ satisfies (24). And the latter in turn requires (16), because $\epsilon > 1$.

$$
1 < \epsilon < \frac{A}{\alpha(1+a)B} \tag{24}
$$

In conclusion, non-negative solutions $w_1, w_2, w_3 \geq 0$ of the inequalities (15) can be constructed as follows. First, I choose a value ϵ that satisfies (24), which exists because of (16). Then, I construct $w_1, w_2, w_3 \geq 0$ backward as in (25). As desired, w_3 is non-negative because the numerator is non-negative by (24) and the denominator is positive because $a^2 \alpha > 1$ by (11).

$$
\begin{aligned}
w_3 &= \frac{1}{2} \frac{A - \epsilon\alpha(a+1)B}{\alpha a^2 - 1} \\
w_2 &= a w_3 + \epsilon B \\
w_1 &= a w_2 + \epsilon B
\end{aligned}
\tag{25}
$$

E Computing the cones

The following reasoning shows that, when the constraints are defined as in (5), the cone $\mathcal{W}(\mathsf{x}_1, \mathsf{y}_1)$

can be described through the inequality $v_1 > av_2$ in (10) with $a = 2.5$.

$$\begin{aligned} & \boldsymbol{v} \in \mathcal{W}(\mathsf{x}_1, \mathsf{y}_1) \\ \iff & \sum_{k=1}^{3} C_k(\mathsf{x}_1, \mathsf{y}_1) v_k < \sum_{k=1}^{3} C_k(\mathsf{x}_1, \mathsf{z}_1) v_k \\ \iff & v_1 > 2.5 v_2 \end{aligned}$$

An analogous reasoning holds for $\mathcal{W}(\mathsf{x}_2, \mathsf{y}_2)$ and $\mathcal{W}(\mathsf{x}_3, \mathsf{y}_3)$.

F Computing the weights

When a and α are chosen as in (19), the coefficients A and B defined as in (17) become $A = 0.540426093542$ and $B = 0.891687359847$. And condition (24) on ϵ becomes (1).

$$(1) \quad 1 < \epsilon < \frac{A}{\alpha(1+a)B} = 1.04947406627$$

Thus, I can choose for instance $\epsilon = 1.03$. The weights in (6) are obtained from (25) with $a = 2.5$, $\alpha = 0.165$, and $\epsilon = 1.03$. These weights thus satisfy the three inequalities (12), (13), and (14).

References

Arto Anttila, Scott Borgeson, and Giorgio Magri. 2019. Equiprobable mappings in weighted constraint grammars. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 125–134. Association for Computational Linguistics.

Arto Anttila and Giorgio Magri. 2018. Does MaxEnt overgenerate? Implicational universals in Maximum Entropy grammar. In *AMP 2017: Proceedings of the 2017 Annual Meeting on Phonology*, Washington, DC. Linguistic Society of America.

Paul Boersma. 1997. How we learn variation, optionality and probability. In *Proceedings of the Institute of Phonetic Sciences (IFA) 21*, pages 43–58, University of Amsterdam. Institute of Phonetic Sciences.

Paul Boersma. 1998. *Functional Phonology*. Ph.D. thesis, University of Amsterdam, The Netherlands. The Hague: Holland Academic Graphics.

Paul Boersma and Joe Pater. 2016. Convergence properties of a gradual learning algorithm for Harmonic Grammar. In John McCarthy and Joe Pater, editors, *Harmonic Grammar and Harmonic Serialism*. Equinox Press, London.

Canaan Breiss and Adam Albright. 2022. Cumulative markedness effects and (non-)linearity in phonotactics. *Glossa: a journal of general linguistics*, 7:1–32.

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a Maximum Entropy model. In *Proceedings of the Stockholm Workshop on Variation Within Optimality Theory*, pages 111–120, Stockholm University.

Bruce Hayes and Aaron Kaplan. 2023. Zero-weighted constraints in Noisy Harmonic Grammar. *Linguistic Inquiry*, pages 1–14.

Bruce Hayes and Colin Wilson. 2008. A Maximum Entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, 39:379–440.

Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky. 1990a. Harmonic Grammar – a formal multi-level connectionist theory of linguistic well-formedness: an application. In *Proceedings of the 12th annual conference of the Cognitive Science Society*, pages 884–891, Hillsdale, NJ. Lawrence Erlbaum Associates.

Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky. 1990b. Harmonic Grammar – a formal multi-level connectionist theory of linguistic well-formedness: theoretical foundations. In *Proceedings of the 12th annual conference of the Cognitive Science Society*, pages 388–395, Hillsdale, NJ. Lawrence Erlbaum.

Giorgio Magri and Arto Anttila. 2023. Paradoxes of MaxEnt markedness. In *AMP 2022: Supplemental Proceedings of the 2022 Annual Meeting on Phonology*. Linguistic Society of America.

Joe Pater. 2009. Weighted constraints in generative linguistics. *Cognitive Science*, 33:999–1035.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality Theory: constraint interaction in generative grammar*. Blackwell, Oxford.

Brian W. Smith and Joe Pater. 2020. French schwa and gradient cumulativity. *Glossa: a journal of general linguistics*, 5:1–33.

Kie Zuraw and Bruce Hayes. 2017. Intersecting constraint families: an argument for Harmonic Grammar. *Language*, 93.3:497–546.

Subject-verb agreement with Seq2Seq transformers: Bigger is better, but still not best

Michael Wilson and **Zhenghao Zhou** and **Robert Frank**
Yale University
370 Temple Street
New Haven, CT 06511
{michael.a.wilson, herbert.zhou, robert.frank}@yale.edu

Abstract

Past work (Linzen et al., 2016; Goldberg, 2019, a.o.) has used the performance of neural network language models on subject-verb agreement to argue that such models possess structure-sensitive grammatical knowledge. We investigate what properties of the model or of the training regimen are implicated in such success in sequence to sequence transformer models that use the T5 architecture (Raffel et al., 2019; Tay et al., 2021). We find that larger models exhibit improved performance, especially in sentences with singular subjects. We also find that larger pre-training datasets are generally associated with higher performance, though models trained with less complex language (e.g., CHILDES, Simple English Wikipedia) can show more errors when trained with larger datasets. Finally, we show that a model's ability to replicate psycholinguistic results does not correspondingly improve with more parameters or more training data: none of the models we study displays a fully convincing replication of the hierarchically-informed pattern of agreement behavior observed in human experiments.

1 Introduction

In standard English, subjects and present-tense verbs covary in number, called *subject-verb agreement*. Crucially, agreement depends not on linear proximity to the verb, but structural proximity: the head noun of the subject determines correct agreement, not any of its dependents:

(1) a. The label on the bottle is...
b. * The labels on the bottle is...
c. The labels on the bottle are...
d. * The label on the bottles are...

Because of this structure-sensitive property of subject-verb agreement, this phenomenon is a useful grounds for examining the linguistic representations that computational language models learn.

Past work examining the performance of language models on subject-verb agreement has found mixed results. Linzen et al. (2016) and Marvin and Linzen (2018) showed LSTMs do not achieve consistent structure-sensitive generalization on agreement when trained on a language modeling task, though they perform better with explicit supervision related to agreement. Goldberg (2019) examined BERT, an encoder-only transformer model (Devlin et al., 2018), and found much higher subject-verb agreement performance.

These prior studies compared language model probabilities for individual word tokens (e.g., *is* vs. *are*) following a preamble (e.g., *the label on the bottles*) to determine whether singular or plural agreement is more likely. We use a different approach, studying agreement in models trained to map an input (non-agreeing) sequence to an output (agreeing) sequence. This follows a line of work in which grammatical transformation tasks can be used to assess sensitivity to grammatical regularities (McCoy et al., 2020; Mueller et al., 2022; Mulligan et al., 2021). Specifically, we use ablations of the Text-to-Text Transfer Transformer (T5) sequence to sequence (seq2seq) architecture (Raffel et al., 2019; Tay et al., 2021) to examine the effect of model size (number of parameters) and model architecture (where those parameters are located) on agreement behavior. As we shall see, bigger models do better, but some kinds of layers matter more for performance. We also investigate how pre-training data influences model performance, examining T5 models that were pre-trained on different datasets and different amounts of data.

Previous work has demonstrated that pre-training imparts a bias to make use of hierarchical generalizations in at least some seq2seq models on tasks like passivization and question formation in English and German (Mueller et al., 2022). Like these tasks, subject-verb agreement is sensitive to hierarchy and not linear order, as shown in (1).

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 278-288.
Amherst, Massachusetts, June 15-17, 2023

However, unlike passivization and question formation, agreement is not a generalization based on movement.[1] This could potentially impact the models' propensity to form hierarchical generalizations in this domain. Indeed, we find that even though the overall propensity to use grammatical agreement increases with model size, even the largest models we tested showed errors. Moreover, the pattern of these errors does not match patterns of errors found in psycholinguistic studies of agreement errors in humans. People show more sensitivity to structural proximity when making errors, while the models we tested showed more sensitivity to linear proximity. We conclude that the most reliable way to achieve higher performance on agreement in general is with larger models, though even the largest models we tested still do not replicate most human-like patterns of agreement errors, and thus show more evidence of linear rather than hierarchical generalization, at least with regards to agreement behavior.

We note here that we do not have a full explanation of why certain architectural properties and kinds of pre-training data have certain affects on agreement behavior. Rather, our more modest aim is merely to provide a sketch of the empirical landscape in this domain.

2 Methods

2.1 Procedure

Sequence to sequence (seq2seq) language models take a sequence of (tokenized) words as input, and produce a sequence of tokens as output. The model begins generation by producing a beginning of sentence token, and then produces the next most probable token at each generation step given the full input sequence and the previous tokens generated in the output sequence to that point.

To assess agreement behavior in these models, we take advantage of the fact that in English, verbs in the past tense are not marked for number (with the single exception of *was* vs. *were*, which was not included in our test set). Thus, we fine-tune the T5 checkpoints we use on a tense reinflection task (McCoy et al., 2020; Mueller et al., 2022; Petty and Frank, 2021; Mulligan et al., 2021). For example:

Source: "The professor liked the dean. PRES: "

Target: "The professor likes the dean."

This task requires the model to convert a sentence where number agreement is absent (i.e., the past tense) to a form where agreement is clearly marked (the present tense), forcing the model to resolve the ambiguity. We measure which form of the present tense verb the model produces.

We fine-tuned all models for $7,812$ weight updates (976.5 epochs) on this tense reinflection task with a learning rate of 5×10^{-5} and a batch size of 128, following Mueller et al. (2022). We saved 15 evenly-spaced checkpoints throughout fine-tuning to use for evaluation.[2]

2.2 Materials

Our fine-tuning dataset consists of $1,098$ examples constructed from sentences randomly drawn from English Wikipedia (`20200501.en`) using Hugging Face's `datasets` library.[3] We parsed the sentences using a transformer-based dependency parser provided by the `spacy` library (`en_core_web_trf`) (Honnibal et al., 2020). These parses allow us to identify the subject of the sentence and the verb, as well as the verb's tense. We created pairs of sentences for fine-tuning as follows: if the verb is in past tense, we treat the sentence as the input, and reinflect the verb into the present tense to produce the desired output; if the verb is in the present tense, we treat it as the desired output, and reinflect it into the past tense to produce the input. For reinflection, we used the `pattern` library (Smedt and Daelemans, 2012), with additional manual corrections. We included only examples that contained no intervening nouns between the main subject and the main verb according to the dependency parses, in order to avoid giving the models evidence during fine-tuning that would disambiguate the correct target of agreement, even inadvertently.[4]

For our test dataset, we created a balanced set of synthetic past-present example pairs using a PCFG. Using synthetic test data allowed us to ensure full

[1] That is, it is a relation that holds between elements in a structure, rather than a relation between structures (as movement is typically defined).

[2] Our code and data are available at: github.com/clay-lab/seq2seq-agreement-attraction-datasets, github.com/clay-lab/seq2seq-agreement-attraction.

[3] We also conducted fine-tuning with larger datasets, up to $10,000$ sentence pairs. Preliminary investigations showed little difference between the results with these larger fine-tuning datasets and the smaller dataset, so we continued to use the smaller dataset.

[4] Preliminary investigations showed that including sentences with interveners where the correct target of agreement was ambiguous in the pre-training data (e.g., *the key to the cabinet is...* is compatible with either a hierarchical or a linear generalization) made little difference to our results.

accuracy of the target forms during testing, since naturally occurring data may contain errors that arise naturally or during parsing. We represent conditions using "S" and "P," with "S" corresponding to a singular noun and "P" corresponding to a plural noun. The linear order of these labels represents their relative linear order in the sentence prior to the verb. For instance, the following is a sentence in the SP condition:

(2) The $\text{students}_{\text{S}}$ near the deans_{P} liked the professor.

Distractor nouns were embedded in either a prepositional phrase (PP) or a subject relative clause (RC), or a combination of two of them, attached to the preceding noun. Thus, there were test sentences for each combination of noun numbers (S, P, SS, SP, PP, PS, SSP, SPS, SPP, PPS, PSP, PSS) and embedding structure (PP, RC (two-noun conditions), PP+PP, PP+RC, RC+RC, RC+PP (three-noun conditions)). The test sentences used 10 nouns in singular and plural forms (*student*, *professor*, *headmaster*, *friend*, *assistant*, *dean*, *advisor*, *colleague*, *president*, *chancellor*), 10 verbs in past and present tense forms (*help*, *visit*, *like*, *bother*, *inspire*, *recruit*, *assist*, *confound*, *accost*, *avoid*), 5 prepositions (*of*, *near*, *by*, *behind*, *with*), the definite article (*the*), and the overt complementizer (*that*). Due to the limited vocabulary and structural simplicity, the S and P conditions each contained only 64 unique sentences each. All other conditions contained 256 unique sentences.

We did not ensure that every sentence had a completely plausible meaning. This is similar to Lasri et al. (2022)'s approach, who examined BERT's performance on subject-verb agreement in sentences without sensible meanings. It is also similar to Newman et al. (2021), who examined how plausibility of a verb in a particular context influenced BERT's ability to predict the syntactically correct form of an agreeing verb. Both studies found that implausible carrier sentences and less plausible verbs in a particular context were associated with a higher rate of errors. While we did not explicitly manipulate plausibility, our results can be similarly interpreted as reflecting models' performance in less than completely natural contexts.

2.3 Evaluation

During preliminary investigations with unconstrained generation of output, we found that the seq2seq models we used often failed to produce output that could be used to determine whether they displayed agreement errors straightforwardly. This was because the models either failed to produce the correct preamble (i.e., the string prior to the main verb); failed to reinflect the verb, leaving it in the past tense; or produced the wrong verb, which made it impossible to parse the output with the CFG used for analysis. For this reason, we used teacher forcing to make the models produce an identical preamble up to the main verb, and then forced them to produce either the singular or plural present tense form of the target verb.[5] This ensures that every output sentence provides information about the model's behavior with regards to agreement, since the output inevtiably reveals whether the model considers the singular or the plural form of the verb more likely given the correct preamble. We ignore the remainder of the output following the main verb for evaluation purposes.

Pre-verb noun(s)	Structures
S	–
P	–
SS, SP; PP, PS	PP; RC
SSS, SSP, SPS, SPP; PPP, PPS, PSP, PSS	PP+PP, PP+RC, RC+PP, RC+RC

Table 1: Summary of test set conditions. The correct target of agreement was always the first noun.

For each example in our test dataset, we record whether the model displayed erroneous agreement, defined as producing the singular form of the verb when the correct target is plural, or vice versa. Our plots show the proportion of errors on the y-axis; thus, higher numbers represent worse performance and lower numbers represent better performance. For each model, we consider results for only the checkpoint that showed the lowest overall proportion of agreement errors.

2.4 Models

We consider several T5 models, drawn from two sources. The first are checkpoints released with Tay et al. (2021), in (3). These models differ in a number of respects with comparison to a "base" model, including the total number of layers (NL), the number of encoder layers (EL), the number of decoder layers (DL), and the number of attention heads (NH).

(3) a. T5 Efficient Tiny, Mini, Small, and Base[6]

[5]This meant that at each generation step, we forced the models to predict only the correct actual token, and used that prediction to feed the next generation step, up to the disambiguating token at the verb.

[6]These models have the following architectures, which vary in several regards relative to T5 Efficient Base. Tiny:

b. Total number of layers (NL): T5 Efficient Base NL02, NL04, NL08, Base (NL12)[7]

c. Number of decoder layers (DL): T5 Efficient Base DL02, DL04, DL06, DL08, Base (DL12)

d. Number of encoder layers (EL): T5 Efficient Base EL02, EL04, EL06, EL08, Base (EL12)

e. Number of attention heads (NH): T5 Efficient Base NH08, Base (NH12), NH16, NH24, NH32

We do not consider other ablations here. This set of models ranges between 16 million parameters on the low end (T5 Efficient Tiny) and 364 million on the high end (T5 Efficient Base NH32). They were all pre-trained on the same dataset drawn from the Colossal Cleaned Common Crawl (C4) corpus, using a span-denoising objective. In total, we considered 19 T5 Efficient models.

To investigate the effects of pre-training data, we used models provided by Aaron Mueller (p.c.). These models each have 63 million parameters, and were pre-trained on a span-denoising objective. Different models were pre-trained on data drawn from different sources, including the CHILDES database (BabyT5), the C4 corpus (C4), Simple English Wikipedia (SimpleWiki), and standard English Wikipedia (WikiT5). The size of the pre-training datasets ranges from 1 million words to 1 billion words, though not every combination of dataset size and source is represented.[8] Altogether, these comprised a separate set of 13 models.

3 Results

3.1 Model size and architecture

First, we consider results for some of the T5 Efficient models (Tay et al., 2021). Figure 1 shows accuracy by condition and number of parameters.

For this and all future statistical results we report, we fit logistic regressions using R's `glm` function (R Core Team, 2022). Throughout the paper, for each family of hypothesis tests, we used the Bonferroni method to correct for multiple comparisons. As shown in (1), performance in most conditions was significantly affected by model size, such that more parameters led to a decreased error rate. The exceptions to this were the single-noun conditions, the PPS PP+PP condition, the PPS RC+PP condition, the PSS PP+PP condition, and the PSS PP+RC condition. In all cases, this appears to be due to the fact that even models with the smallest number of parameters we considered achieved high performance in these conditions, leaving little to no room for further improvement.

We next consider which kinds of parameters have effects. Naturally, increasing the number of layers (for example) increases the number of parameters. But we can also consider whether increasing the number of attention heads without increasing the number of layers is beneficial. Figure 2 shows the overall proportion of errors for the number of encoder layers, decoder layers, total layers, and attention heads per layer.

Both increasing the number of layers, as well as the number of attention heads per layer, significantly improves model performance (NL: $\beta = -0.0780$, $z = -83.9$, $p < 2.2 \times 10^{-16}$; NH: $\beta = -0.0870$, $z = -63.1$, $p < 2.2 \times 10^{-16}$). In addition, increases in the number of encoder layers and in the number of decoder layers both improve performance as well (EL: $\beta = -0.0948$, $z = -64.8$, $p < 2.2 \times 10^{-16}$; DL: $\beta = -0.101$, $z = -68.7$, $p < 2.2 \times 10^{-16}$). We found, however, that increasing the number of encoder layers resulted in a significantly greater increase in performance compared to increasing the number of decoder layers (EL $-$ DL: $\beta = -0.00593$, $z = -2.87$, $p = 0.00415$). The negative slope for the difference indicates that the magnitude of the EL effect is greater than the magnitude of the DL effect. Thus, assigning more parameters to encoding layers when increasing model size appears to carry a greater benefit with regards to overall agreement behavior in our test dataset.

This effect could in principle have two sources. One possibility is obvious: increasing the number of encoder layers provides greater benefits with regards to our tense-reinflection task and/or subject-verb agreement. But another possibility is that models with fewer decoder layers show less reduction in performance compared to models with more decoder layers, leaving less room for improvement as the number of decoder layers is increased. To investigate this, we can compare the intercepts of the regressions. We found that the intercept for the encoder-layer model was -0.661, while the intercept for the decoder-layer model was -0.618. This

NL04 (EL04, DL04), NH04; Mini: NL04 (EL04, DL04), NH08; Small: NL06 (EL06, DL06), NH08; Base: NL12 (EL12, DL12), NH12.

[7] Using the convention from Tay et al. (2021), the number by "NL" signifies half the total number of layers; e.g., NL02 means there are 2 encoder layers and 2 decoder layers (4 total).

[8] BabyT5: 1M, 5M; C4: 1M, 10M, 100M, 1B; SimpleWiki: 1M, 10M, 100M; WikiT5: 1M, 10M, 100M, 1B.

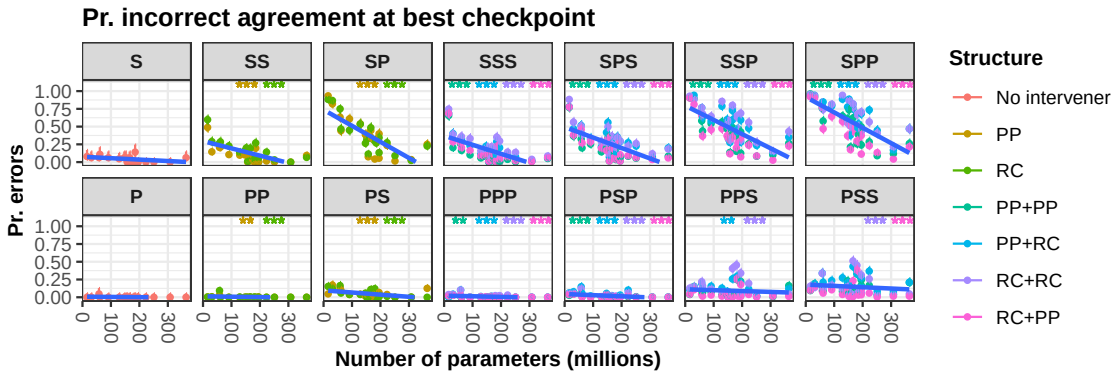


Figure 1: Accuracy by number of parameters and condition. Bars represent 95% CIs on the beta distribution. Colored stars indicate significance of the corresponding condition with Bonferroni-corrected α.

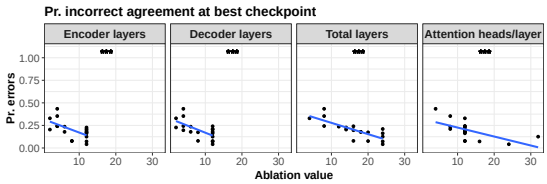


Figure 2: Accuracy by ablation type. Stars indicate significance with Bonferroni-corrected α.

indicates that the models with fewer encoder layers are *less* likely to make errors than the models with fewer decoder layers, and this difference is significant (EL − DL intercept: $\beta = 0.0435$, $z = 2.53$, $p = 0.01$). Thus, we find evidence that the difference reflects a genuine advantage for increased number of encoder layers on our task. We have no ready explanation for why this should be (in principle, agreement could be determined in either the encoder or the decoder, or equally in both). Nevertheless, we find this result interesting given the current focus of the field on decoder-only models like LLaMA (Touvron et al., 2023) and GPT (Brown et al., 2020; OpenAI, 2023). Our results suggest that for some tasks, it may be possible to more efficiently achieve higher performance with a model that incorporates an encoder.

When considering effects of this sort by condition (which we do not plot), we again found that in most conditions, increases in the relevant number of layers/heads led to improved performance. However, there were exceptions, summarized in table 2. In all other conditions, there were improvements in performance associated with increasing the parameters of each type. The single clear pattern is that performance in the plural subject conditions is less often improved by increasing model size, again likely due to the low error rate in these conditions to begin with. It is unclear to us why this should be the case; we recorded the number of singular and

Ablation(s)	Noun(s)	Structure(s)
DL, TL	S	–
EL, DL, TL, NH	P	–
NH	PP	PP, RC
NH	PS	PP
EL, DL	PPP	PP+PP
NH	PPP	PP+RC
EL, DL, TL, NH	PPS	PP+PP
DL, NH	PPS	PP+RC
EL, DL, TL	PPS	RC+PP
TL	PPS	RC+RC
EL, DL, TL, NH	PSS	PP+PP
DL, TL, NH	PSS	PP+RC
DL	PSS	RC+PP, RC+RC

Table 2: Summary of conditions where no improvement associated with various ablations was found.

plural subjects in our fine-tuning data and found that 89% of subjects were singular, while 11% were plural, which if anything should be expected to produce higher accuracy in the singular subject conditions. For instance, if the model simply assigns higher probability to the more frequent form, this should be correct most of the time in the singular-subject condition. One possibility (suggested by a reviewer) is that when there are conflicting signals about agreement, the models default to the morphologically unmarked plural form.

Another possibility is that this behavior is due to an artifact of how the models tokenize certain verbs we used in our test set. In some cases, the models tokenize a singular verb as two tokens (e.g., *like* and *s* for *likes*). Due to how we used teacher-forcing, this meant that the models were forced to predict identical tokens up until the disambiguating token, which for a word like *like(s)* would be the token following *like*. After this, the models were forced to predict either the singular continuation, *s*, or a token that was the beginning of a word (indicated in the sentence piece tokenizer as tokens that begin with a special unicode character). This

regimen may have masked cases where the models predictions were poor before the verb, leading the model to enter a state where it was being forced to choose the best continuation for a sequence that it considers low probability to begin with. In this case, the following token may have been chosen erroneously, but in the plural conditions, this would still look like the model had correctly predicted the plural verb. Distinguishing between the possibilities will require further investigation.

3.2 Amount and kind of pre-training data

We next consider the effects of pre-training data on agreement behavior while holding model size constant. We consider T5 models with 63M parameters, pre-trained on CHILDES (MacWhinney, 2000), Simple English Wikipedia (simple.wikipedia.org), English Wikipedia (en.wikipedia.org), and C4 (Raffel et al., 2019). Figure 3 shows the proportion of errors by dataset type and size for each condition.

Due to the limited number of models we had available for each source of pre-training data (2 for CHILDES, 3 for Simple English Wikipedia, and 4 each for C4 and English Wikipedia), we classified models as having been pre-trained on either simple English (CHILDES, Simple English Wikipedia) or standard English (C4, English Wikipedia). We fit logistic regressions using the `glmer` function from R's `lme4` library (Bates et al., 2015) with random intercepts and slopes for each individual source of data, with p-values obtained using the `lmerTest` library (Kuznetsova et al., 2017). To address statistical concerns, we used the $\log_{10}$ of the dataset size in words as a predictor.

When predicting errors across all conditions, we found a significant main effect of dataset size ($\beta = -0.18633$, $z = -6.979$, $p < 0.001$), indicating improved performance as the size of the pre-training dataset increases. However, there was no effect of language complexity (i.e., simple vs. standard English) ($\beta = -0.06758$, $z = -0.375$, $p = 0.707$), nor any interaction between complexity and size ($\beta = 0.01620$, $z = 0.465$, $p = 0.642$).

As before, the effect of dataset size was significant in most conditions for both types of models. However, as (3) shows, for models pre-trained on simple English, more data led to a higher error rate in the SP, SSP, and SPP conditions. In contrast, for models pre-trained on standard English, all effects found went in the expected direction. We would urge caution in over-interpreting these results, since even the largest of the datasets we consider here, at 1 billion words, is much smaller than the C4 dataset used to pre-train the T5 Efficient models we consider earlier, which consists of approximately 156 billion tokens (Dodge et al., 2021). While the unit of measurement used to report the size of these datasets differs, it seems clear that the full C4 corpus is roughly 100 times larger than the largest dataset used to pre-train these models. A fuller study of properties of the different corpora used may shed light on this behavior, though this is beyond the scope of this paper.

Nevertheless, we find it interesting that in some cases larger datasets led to increased errors, which may be due to a kind of overfitting to the simpler data that made the models less robust to longer sentences with multiple nouns prior to the main verb. However, notably, these conditions all have singular subjects and plural interveners, which is known to lead to increased agreement errors in people. This leads us to a consideration of whether the kinds of agreement errors the models make are in general like those people make.

3.3 Agreement attraction

Psycholinguistic studies have found some linguistic contexts lead to more agreement errors than others. A common feature of contexts that lead to more of these errors is the presence of a noun that linearly intervenes between the head noun of the subject (the correct target) and the verb that has a different number feature from the correct target. This is a feature in most of our conditions. For example, more agreement errors are produced after preambles like (4b) than after preambles like (4a) (Bock and Cutting, 1992).

(4) a. The key to the cabinet...
b. The key to the cabinets...

Intuitively, the reason (4b) prompts more errors than (4a) is due to the plural noun, *cabinets*. The noun interferes with the correct target of agreement, *key*, leading to increased production of an incorrect plural verb. This kind of error is referred to as *agreement attraction*.

Recent work has examined to what extent language models replicate patterns of human language use (e.g., Arehalli and Linzen, 2020; Brennan et al., 2020; Hao et al., 2020; Wilcox et al., 2021). It is possible the errors of the models we investigate reflect a human-like understanding of agreement. This could be true if errors are disproportionately

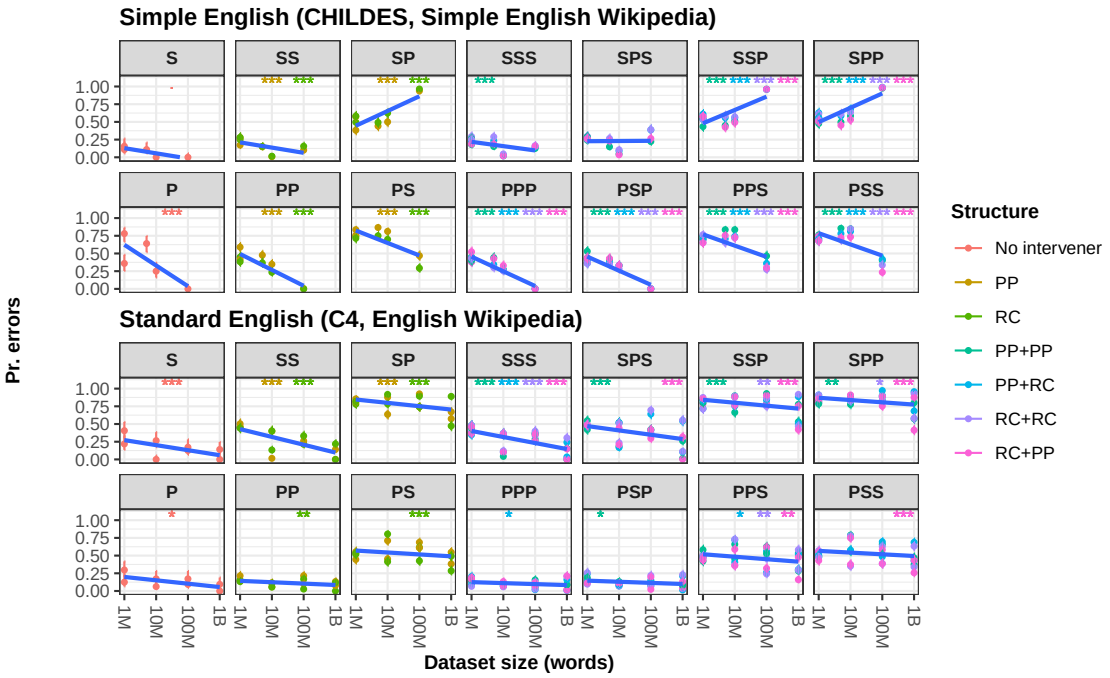


Figure 3: Accuracy by dataset size, type, and condition at each model's best overall checkpoint. Colored stars indicate significance of the corresponding condition with Bonferroni-corrected α.

concentrated in contexts where people make relatively more agreement errors. Arehalli and Linzen (2020) investigated this question with LSTMs pretrained on English Wikipedia. They used preambles taken from psycholinguistic studies of agreement attraction, and measured the models' predictions for *is* or *are* as the following token. Their models replicated some but not all agreement attraction effects. Like people, their LSTMs showed more attraction for distractors in PPs than distractors in RCs, effects of adjacency in coordinate structures, and sensitivity to clause-external distractors. However, unlike people, they were more influenced by linear adjacency than structural proximity, and showed no effect of notional number nor of argument vs. adjunct status of the distractor. We examine the singular-plural asymmetry, structure (PP vs. RC) and linear adjacency (e.g., SPS vs. SSP) to determine how similarly the T5 models we tested behave compared to people.

3.3.1 Singular-plural asymmetry

Bock and Cutting (1992) found that people produce more agreement errors after (4b) than after (5).

(5) The keys to the cabinet...

In other words, more errors arise with singular subjects and plural interveners (SP) than with plural subjects and singular interveners (PS).

Figure 4 shows the difference in the proportion of agreement errors for the SP and PS conditions by model. A positive value indicates more errors in SP than in the PS conditions, and thus a singular-plural asymmetry that goes in the same direction as observed in psycholinguistic experiments.

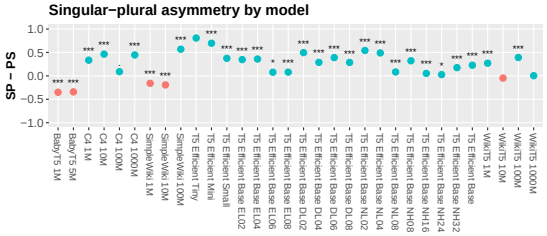


Figure 4: Singular-plural asymmetry by model for the two-noun conditions. Stars indicate significance obtained from χ^2 tests comparing accuracy across the two conditions with Bonferroni-corrected α.

Most models show the same asymmetry as people; the exceptions are BabyT5, SimpleWiki 1M and 10M, and WikiT5 10M (with only the latter difference not statistically significant). The overall pattern is not so surprising given fig. (1), but this shows the differences by model.

3.3.2 Structural context of distractor

In addition to the morphologically-based singular-plural asymmery, Bock and Cutting (1992) also

showed that people were more likely to make errors when the intervener was embedded in a PP (6a) compared to when it was embedded in an RC (6b), a structural asymmetry.

(6) a. The student in the classes...

b. The student who failed the classes...

Figure 5 shows the difference in the proportion of agreement errors for the PP and RC two-noun conditions. A positive value indicates more errors in the PP conditions than in the RC conditions, and thus a PP-RC asymmetry that matches the results of Bock and Cutting (1992).

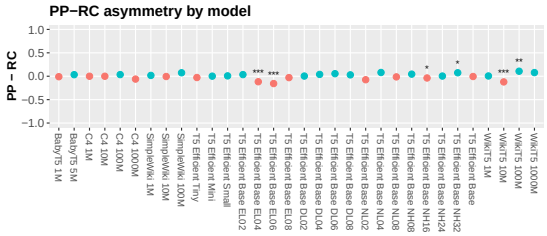


Figure 5: PP-RC asymmetry by model for the two-noun conditions. Stars indicate significance obtained from χ^2 tests with Bonferroni-corrected α.

In this case, 12 of the 32 models showed a numerical asymmetry in the opposite direction compared to people.Of these, only the differences for T5 Efficient Base EL04, EL06, and NH24; and WikiT5 10M are statistically significant. Even for those models with the expected asymmetry, it is less pronounced than the singular-plural asymmetry is in most models, with only two models showing a significant difference in the expected direction (T5 Efficient Base NH32 and WikiT5 100M).

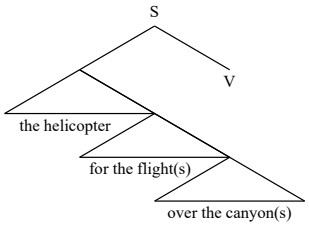
3.3.3 Linear vs. structural proximity

People are more likely to produce agreement attraction errors for distractors that are structurally closer to the verb compared to distractors that are linearly closer but structurally more distant. Franck et al. (2002) found that preambles like (7a) led to more errors than preambles like (7b).

(7) a. The helicopter for the flights over the canyon...

b. The helicopter for the flight over the canyons...

c.



As shown in (7c), the noun that mismatches the subject in number is structurally closer to the verb in (7a) than in (7b). Figure 6 shows three asymmetries that are relevant to this question.

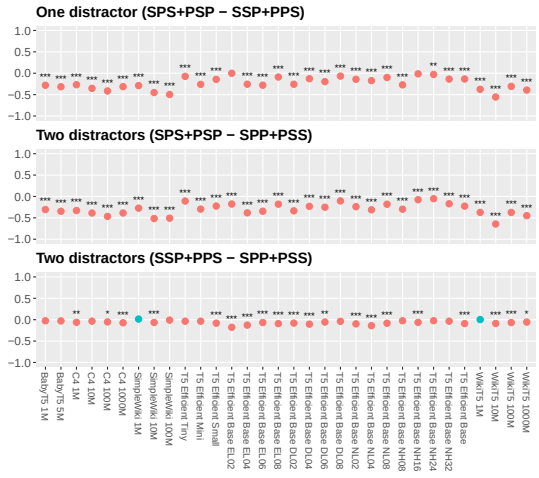


Figure 6: Comparison of multiple-distractor conditions. Stars indicate significance obtained from χ^2 tests comparing accuracy across the two conditions with Bonferroni-corrected α.

The top plot shows the accuracy difference between structural vs. linear closeness for the single-distractor conditions, e.g., SPS (structurally close) and SSP (linearly close). All differences are < 0, indicating that the models' performance is worse when distractors are linearly closer to the verb, with differences for all but two models (T5 Efficient Base EL02 and NH16) being statistically significant. The middle plot shows the difference between the conditions with a single structurally close distractor (e.g., SPS) and conditions with structurally and linearly close distractors (e.g., SPP). Though the SPP and PSS conditions contain structurally and linearly close distractors, Franck et al. (2002) found attraction errors were highest in the single, structurally close distractor conditions, such that, e.g., SPS led to more errors than SPP. The models fail to replicate this pattern, showing worse performance in the multiple distractor conditions than in the single distractor conditions, since all differences are < 0. All of these differences are statistically significant. Finally, the lowest row shows the difference between the single, linearly close distractor conditions and the multiple distractor conditions. A negative value means that the model shows more attraction with two distractors compared to one, unlike Franck et al. (2002)'s results. Nearly all of the models behave this way; the sole exceptions are SimpleWiki 1M and WikiT5 1M. However, the

negative differences for BabyT5 1M and 5M; C4 10M; SimpleWiki 100M; T5 Efficient Tiny, Mini, Base DL08, Base NH08, Base NH24, and Base NH32 are not statistically significant; neither of the positive differences are statistically significant.

In general, unlike what Franck et al. (2002) found, the models are more likely to make attraction errors when distractors are linearly adjacent to the verb compared to when they are structurally adjacent, and they are more likely to make errors when there are multiple distractors that intervene between the subject and the main verb.

A potential confound is that the locus of attachment may be ambiguous in our synthetic data. While Franck et al. (2002) controlled for this by word choice (as shown in (7c), where the alternative "high-attachment" parse of the final modifier would be semantically anomalous), our synthetic test dataset did not. As such, the "correct" parse of the three-noun conditions is potentially ambiguous. Nevertheless, due to how our PCFG was defined, high-and low-attachment parses of the final modifier should be equally plausible. Despite this, we still found significant differences for most models when the distractor was linearly adjacent to the the verb, and when there were multiple distractors. This suggests to us that the models' performance is typically significantly influenced by linear adjacency, since we might have otherwise expected at worst chance performance. Furthermore, Franck et al. (2002) found that for people, there was little difference between the single, structurally close distractor conditions (e.g., SPS and PSP) and the multiple distractor conditions (e.g., SPP and PSS), while the models show significantly higher error rates with multiple distractors. Thus, despite the potential ambiguity, most models behave consistently differently from people in this regard.[9]

4 Conclusion

We examined pre-trained T5 models to determine how model size, architecture, dataset size, and dataset type affected subject-verb agreement on a tense reinflection task. We found that bigger models performed better, especially in singular-subject conditions. In contrast, model performance was already high even for small models in the plural-subject conditions. Increasing the number of layers as well as the number of attention heads per layer result in improvements, though adding encoder layers was associated with greater improvement than adding decoder layers.

When considering the type and amount of pre-training data, we found increasing the amount of pre-training data improved agreement accuracy overall. However, for the models trained on simple English text (CHILDES, Simple English Wikipedia), bigger training datasets led to **worse** performance in singular-subject conditions with linearly-adjacent distractors (e.g., SP, SSP, SPP), despite leading to better performance in plural subject conditions. In contrast, for models trained on standard English (C4, English Wikipedia), more pre-training data uniformly led to increased performance (when performance with small datasets was not already high).

The models did not consistently display patterns reminiscent of agreement attraction. While most models showed a number asymmetry matching what has been found in psycholinguistic work, other asymmetries found in agreement attraction errors were not present. Unlike the LSTMs examined in Arehalli and Linzen (2020) and unlike the results of Bock and Cutting (1992), only some of the transformer models we considered produced more errors in PP than in RC conditions. However, similarly to Arehalli and Linzen (2020)'s LSTMs, the transformer models still showed more attraction for linearly adjacent distractors compared to structurally closer distractors, in addition to showing worse performance with multiple distractors.

Our results show both the advantages and limitations of increasing the size of models and datasets. While increases in both of these independently lead to better performance on subject-verb agreement, an indirect indicator of hierarchical knowledge of language, not even the largest models we considered, nor those pre-trained on the largest amounts of data, display fully human-like behavior. Instead, they were still susceptible to linear interference to a much greater degree than people are (cf. Petty and Frank, 2021). It appears these perennial issues of hierarchical vs. linear generalization with regards to language modeling remain a concern for transformers even now.

[9] We have also conducted preliminary investigations on the models' performance using a span-denoising task on the actual stimuli used in Franck et al. (2002), and found that even on those stimuli, the models display essentially the same sensitivity to linear over structural proximity, though we have not yet conducted statistical tests.

Acknowledgments

We would like to thank Aaron Mueller for sharing his T5 models with us. We also thank the members of the Computational Linguistics at Yale lab and three anonymous reviewers for suggestions and feedback. This work was made possible by support from the National Science Foundation grant BCS-1919321.

References

Suhas Arehalli and Tal Linzen. 2020. Neural language models capture some, but not all, agreement attraction effects. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*, pages 370–376.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Kathryn Bock and J. Cooper Cutting. 1992. Regulating mental energy: Performance units in language production. *Journal of Memory and Language*, 31:99–127.

Jonathan R. Brennan, Chris Dyer, Adhiguna Kuncoro, and John T. Hale. 2020. Localizing syntactic predictions using recurrent neural network grammars. *Neuropsychologia*, 146.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Julie Franck, Gabriella Vigliocco, and Janet Nicol. 2002. Subject-verb agreement errors in French and English: The role of syntactic hierarchy. *Language and Cognitive Processes*, 17(4):371–404.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *CoRR*, abs/1901.05287.

Yiding Hao, Simon Mendelsohn, Rachel Sterneck, Randi Martinez, and Robert Frank. 2020. Probabilistic predictions of people perusing: Evaluating metrics of language model performance for psycholinguistic modeling. In *Proceedings of the Cognitive Modeling and Computational Linguistics (CMCL) Workshop (EMNLP)*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in Python.

Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. 2017. lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13):1–26.

Karim Lasri, Alessandro Lenci, and Thierry Poibeau. 2022. Does BERT really agree ? fine-grained analysis of lexical dependence on a syntactic task. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2309–2315, Dublin, Ireland. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*, third edition, volume II: The Database. Lawrence Erlbaum Associates, Mahwah, NJ.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. *CoRR*, abs/1808.09031.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? Sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. 2022. Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1352–1368, Dublin, Ireland. Association for Computational Linguistics.

Karl Mulligan, Robert Frank, and Tal Linzen. 2021. Structure here, bias there: Hierarchical generalization by jointly learning syntactic transformations. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 125–135, Online. Association for Computational Linguistics.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021. Refining targeted syntactic evaluation of language models. In *Proceedings of the 2021 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 3710–3723, Online. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Jackson Petty and Robert Frank. 2021. Transformers generalize linearly. *CoRR*, abs/2109.12036.

R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *Journal of Machine Learning Research*, 13(66):2063–2067.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2021. Scale efficiently: Insights from pre-training and fine-tuning transformers. *CoRR*, abs/2109.10686.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ethan Wilcox, Pranali Vani, and Roger Levy. 2021. A targeted assessment of incremental processing in neural language models and humans. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 939–952, Online. Association for Computational Linguistics.

Subregular Tree Transductions, Movement, Copies, Traces, and the Ban on Improper Movement

Thomas Graf
Stony Brook University
Department of Linguistics
100 Nicolls Road, Stony Brook, NY 11794, USA
mail@thomasgraf.net

Abstract

Extending prior work in Graf (2018, 2020, 2022c), I show that movement is tier-based strictly local (TSL) even if one analyzes it as a transformation, i.e. a tree transduction from derivation trees to output trees. I define *input strictly local (ISL) tree-to-tree transductions with (lexical) TSL tests* as a tier-based extension of ISL tree-to-tree transductions. TSL tests allow us to attach each mover to all its landing sites. In general, this class of transductions fails to attach each mover to its final landing site to the exclusion of all its intermediate landing sites, which is crucial for producing output trees with the correct string yield. The problem is avoided, though, if syntax enforces a variant of the Ban on Improper Movement. Subregular complexity thus provides a novel motivation for core restrictions on movement while also shedding new light on the choice between copies and traces in syntax.

1 Introduction

Subregular syntax (Graf, 2018; Graf and De Santo, 2019) is a recent research program that explores whether syntactic dependencies, when modeled over suitable representations, fall within very restricted classes in the subregular hierarchy of formal (string or tree) languages. The program has many parallels to *subregular phonology* (see Heinz 2018 and references therein), which has shown that phonology is very restricted in its expressivity: I) well-formedness conditions in phonology are *strictly local* (SL), *tier-based strictly local* (TSL) (Heinz et al., 2011; McMullin, 2016), or some natural extension of TSL (Graf and Mayer, 2018; Mayer and Major, 2018; De Santo and Graf, 2019), and II) a large number of phonological mappings from underlying representations to surface forms are *input strictly local* (ISL) (Chandlee, 2014; Chandlee and Heinz, 2018), with only some falling into more complex classes (Jardine, 2016; Heinz, 2018). The limited nature of phonology furnishes new learning algorithms and novel explanations of typological gaps, and subregular syntax seeks to replicate this success for syntax.

A lot of attention in subregular syntax has been devoted to the operations *Merge* and *Move* in Minimalist syntax and Minimalist grammars (Stabler, 1997, 2011). Merge establishes head-argument relations, whereas Move relates a subtree to multiple positions in the structure. Graf (2018) showed that the constraints that regulate the application of Merge and Move in the syntactic derivation are SL for Merge and TSL for Move, which mirrors the central role of these two classes in phonology. But Merge and Move are structure-building operations and thus inherently transductive: a syntactic derivation is translated into a specific output structure. Recently, the ISL string transductions from subregular phonology have been generalized to trees (Graf, 2020; Ji and Heinz, 2020; Ikawa et al., 2020), and it is fairly easy to see that Merge can be construed as an ISL tree transduction.[1] However, ISL tree transductions cannot handle the long-distance dependencies induced by Move (the long-distance nature of Move is also why the constraints on Move are TSL but not SL). An upper complexity bound on Move exists in the form of deterministic multi bottom-up tree transductions (Kobele et al., 2007), but a tighter, subregular bound remains to be established.

This paper provides a subregular class of transductions for Move by enriching (deterministic) ISL tree-to-tree transductions with a specific TSL mech-

[1] The three generalizations in Graf (2020), Ji and Heinz (2020) and Ikawa et al. (2020) are all distinct and probably incomparable. Graf (2020) generalizes the context-based definition of ISL in Chandlee and Heinz (2018), Ji and Heinz (2020) takes as their vantage point the finite-state machine definition of ISL in Chandlee (2014), and Ikawa et al. (2020) starts with the logic-based perspective of ISL string transductions. Despite these differences, all three can handle the mapping from dependency trees to phrase structure trees *modulo* movement. For the rest of the paper, I will use the term *ISL tree transductions* to refer to the specific version defined in Graf (2020).

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 289-299.
Amherst, Massachusetts, June 15-17, 2023

anism that makes it possible to attach movers to their landing sites. This is sufficient to implement a copy-based version of movement, which is commonly assumed in Minimalist syntax. Producing a structure with the correct string yield, however, requires the ability to distinguish final landing sites from intermediate ones so that movers can be attached only to the former while the latter are filled with traces. The extended version of ISL tree transductions in this paper cannot draw this distinction in the general case, but it is possible in the special case where the distinction is lexically inferrable (in subregular terms, it is SL-1): given a mover m with a set $S := \{f_1, \ldots, f_n\}$ of features that tell us which movement steps m undergoes, inspection of S is sufficient to determine which f_i is the final movement step. This is a relaxed variant of the Ban on Improper Movement (BoIM), and I conjecture that this *output-oriented BoIM* is satisfied in all natural languages.

The paper proceeds as follows. The background section in §2 starts with a general overview of the assumed syntactic formalism, in particular feature-annotated lexical items, dependency trees, and tree tiers (§2.1). This is followed in §2.2 by a discussion of the ISL tree-to-tree mappings in Graf (2020), which are then extended with lexical TSL tests in §3 to capture basic cases of movement. As we will see in §4, this is sufficient to attach movers to all their landing sites. But correct linearization requires placing each mover only in its final landing site, which is a harder problem and prompts my conjecture that all languages satisfy the output-oriented BoIM. A few remaining issues with this overall system are discussed in §5. While care has been taken to make the paper as approachable as possible, it necessarily presupposes a certain amount of familiarity with subregular linguistics, in particular subregular syntax. The reader may want to consult Graf (2022a,b) for a less technical introduction.

2 Background

2.1 Features, dependency trees, and tiers

Subregular syntax measures the complexity of syntax not over strings but over specific types of tree representations. Following Graf and Kostyszyn (2021) and Graf (2022c), I take syntactic derivations to be encoded in the form of dependency trees where each node is a feature-annotated lexical item (LI) in the spirit of Minimalist grammars (Stabler, 1997, 2011).

Definition 1 (Lexical item). Every lexical item is a member of $\Sigma \times \text{Sel}^* \times \text{Lcr}^* \times \text{Cat} \times \wp(\text{Lce})$, where Σ is the set of *phonetic exponents*, Sel is the set of *selector features* $\texttt{F}^+$, Lcr is the set of *licensor features* $\texttt{f}^+$, Cat is the set of *category features* $\texttt{F}^-$, and Lce is the set of *licensee features* $\texttt{f}^-$. ⌟

Category and selector features (by convention in upper case) regulate the application of Merge to establish head-argument relations. Licensor and licensee features (in lower case) trigger Move, with licensor features appearing on the target of movement while licensee features mark the phrase that is moving. The order of features on an LI indicates the order of the operations in which it participates. In contrast to standard MGs, licensee features are unordered so that a mover with licensee features $\texttt{f}_1^-, \ldots, \texttt{f}_n^-$ targets, for each $\texttt{f}_i^-$, the closest properly dominating node with $\texttt{f}_i^+$ ($1 \leq i \leq n$). The removal of order for licensee features does not affect weak generative capacity — this is an easy corollary of the single movement normal form theorem for MGs (Graf et al., 2016).[2] To reduce clutter, we omit $\{\}$ for LIs with no licensee features. In line with MG convention, I use a double colon to separate the LI's phonetic exponent from its feature annotation.

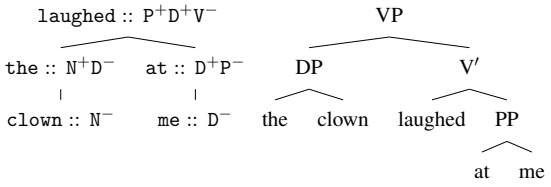
Example. The noun *movie* corresponds to the LI $\texttt{movie} :: \texttt{N}^-$ with phonetic exponent *movie* and category feature $\texttt{N}^-$. The empty T-head — commonly assumed in Minimalist syntax as furnishing the surface position for subjects — is $\varepsilon :: \texttt{V}^+\texttt{nom}^+\texttt{T}^-$. This means that after selecting a VP, the empty T-head provides a landing site for subject movement via $\texttt{nom}^+$, at which point it becomes a full TP that can be taken as an argument by another LI. The LI $'\texttt{s} :: \texttt{N}^+\texttt{D}^+\texttt{D}^- \{\texttt{nom}^-, \texttt{wh}^-\}$ is a possessive marker that takes an NP as its complement, a DP as its specifier, is then selected by another LI with $\texttt{D}^+$, and finally undergoes two movement steps: subject movement via $\texttt{nom}^-$, and wh-movement via $\texttt{wh}^-$. The order of the two movement steps is not fixed and depends on whether the closest properly dominating LI with a matching licensor feature carries $\texttt{nom}^+$ or $\texttt{wh}^+$.

Definition 2 (Dependency tree). Let Lex be a fi-

[2] The definition of LIs above also differs from that of standard MGs in that it does not allow any licensor features to appear before any selector features. This is just a matter of convenience and nothing in this paper hinges on this additional restriction.

nite set of LIs, and $\text{Lex}^{(i)} \subseteq \text{Lex}$ the set of all LIs in Lex with i selector features. The set $\mathbb{D}$ of *(freely combined) dependency trees over* Lex is defined recursively: $l \in \mathbb{D}$ for all $l \in \text{Lex}^{(0)}$, and for all $d_1, \ldots, d_n \in \mathbb{D}$ and $l \in \text{Lex}^{(n)}$, $l(d_n, \ldots, d_1) \in \mathbb{D}$. If m is the mother of node n and n has exactly i right siblings, we say that n is the $(i+1)$-th argument of m. ⌟

Example. A dependency tree for a simple VP is shown below with its corresponding bare phrase structure tree. Each mother-daughter relation in the dependency tree encodes a head-argument relation established via application of Merge.



In general, dependency trees have to satisfy additional linguistic conditions. The root must carry category feature $\texttt{C}^-$, and if m's i-th selector feature is $\texttt{F}^+$, then its i-th argument must carry category feature $\texttt{F}^-$. These constraints regulate the application of Merge and are of little interest for the purposes of this paper. The constrains on Move, on the other hand, merit detailed discussion as they illustrate the use of *tree tiers*.

Definition 3 (Tiers). Let $d \in \mathbb{D}$ be a dependency tree over Lex, and let $T \subseteq \text{Lex}$ be a *tier alphabet*. Given a node x, the predicate $T(x)$ is true iff x is an LI in T. The *T-tier* of d is defined in terms of T-dominance ($\lhd_T^+$), T-mother-of ($\lhd_T$), and T-left-sibling ($\prec_T$), which in turn are expressed in terms of proper dominance in d ($\lhd^+$), reflexive dominance in d ($\lhd^*$), and the left sibling relation in d ($\prec$).

$$
\begin{aligned}
x \lhd_T^+ y \Leftrightarrow & T(x) \wedge T(y) \wedge x \lhd^+ y \\
x \lhd_T y \Leftrightarrow & x \lhd_T^+ y \wedge \neg \exists z [x \lhd_T^+ z \wedge z \lhd_T^+ y] \\
x \prec_T y \Leftrightarrow & \exists z [z \lhd_T x \wedge z \lhd_T y] \wedge \\
& \exists z, z' [z \lhd^* x \wedge z' \lhd^* y \wedge z \prec z']
\end{aligned}
$$

In order to ensure that every tier is a tree, we stipulate that there is a unique node $\rtimes$ such that every node on tier T is either identical to $\rtimes$ or is T-dominated by $\rtimes$. We also stipulate that each leaf is the mother of a distinguished element $\ltimes$. ⌟

Example. The tier alphabet $\texttt{nom}$ of the $\texttt{nom}$-tier contains all LIs with $\texttt{nom}^-$ or $\texttt{nom}^+$, and nothing else. Similarly, the tier alphabet $\texttt{wh}$ of the $\texttt{wh}$-tier contains all and only those LIs that carry $\texttt{wh}^-$ or $\texttt{wh}^+$. The corresponding tier mother-of relations $\lhd_{\texttt{nom}}$ and $\lhd_{\texttt{wh}}$ are shown in Fig. 1 with dashed and dotted lines, respectively, for the dependency tree for *Who said that the clown laughed at me*. As shown in the same figure, these tiers can also be depicted as separate *projections* of the dependency tree.

Intuitively, tiers capture a specific kind of relativized locality (related to but distinct from Rizzi's (1990) notion of Relativized Minimality). If x is the T-mother of y, then x is the closest node that properly dominates y and belongs to a fixed subset T of Lex. For movement, each tier factors out all LIs that are not pertinent to that type of movement. In order for a dependency tree to be well-formed, the following two conditions must hold for every $\texttt{f}$-tier, where $\texttt{f}$ is some movement type ($\texttt{nom}$, $\texttt{wh}$, and so on): I) if x carries $\texttt{f}^-$, then its tier mother carries $\texttt{f}^+$, and II) if x carries $\texttt{f}^+$, exactly one of its tier daughters carries $\texttt{f}^-$.

Mathematically, these conditions are expressed for each tier T via a *licensing function* f_T that maps every $l \in T$ to a string language over T. Tier T is well-formed iff it holds for every node n of T with label l and tier daughters $d_1, \ldots, d_n$ that $d_1 \cdots d_n$ is a string in $f_T(l)$.[3] For example, if l is an LI with $\texttt{f}^+$, then $f_T(l)$ is the set of all strings over T that contain exactly one LI with $\texttt{f}^-$. That every LI with $\texttt{f}^-$ has a tier mother with $\texttt{f}^+$ follows indirectly from the fact that only LIs with $\texttt{f}^+$ may have LIs with $\texttt{f}^-$ in their daughter string.

The complexity of the conditions on Move is measured in terms of the complexity of the string languages used in the licensing functions. A constraint C on a set D of dependency trees over Lex is in the class TSL[TSL] (where TSL is short for *tier-based strictly local*) iff there is some $T \subseteq \text{Lex}$ such that I) f_T maps every $l \in T$ to a TSL-string language in the sense of Heinz et al. 2011 ("the daughter strings are TSL"), and II) for every $d \in D$, C is satisfied in d iff the T-tier of d is well-formed ("C is local over tree tiers"). The two constraints above on movement are TSL[TSL] in this sense (see Graf and Kostyszyn, 2021).

[3] The use of a string-based licensing function is necessary because tree tiers are unranked. There is no upper bound on how many daughters may have, and hence the licensing relations between a mother and its daughters has to be modeled as a licensing relation between a mother and its string of daughters.

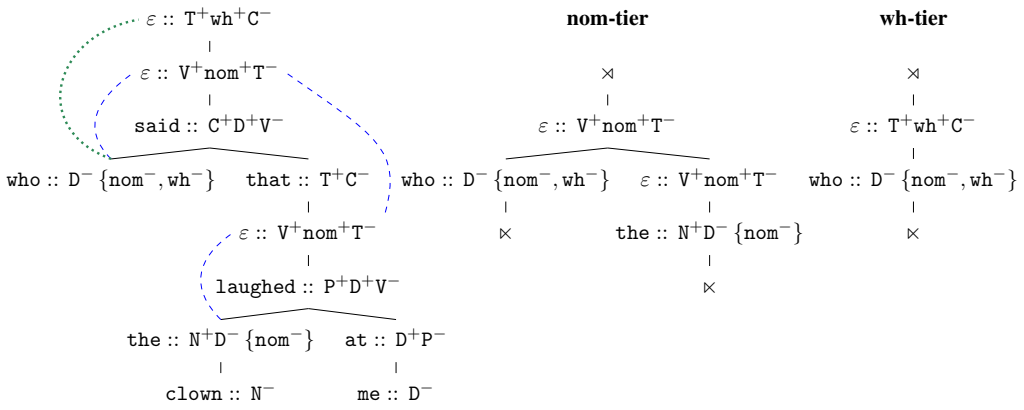


Figure 1: Left: dependency tree for *who said that the clown laughed at me*, with dashed lines representing $\lhd_{\text{nom}}$ and dotted lines representing $\lhd_{\text{wh}}$; Middle and Right: corresponding depictions as tree tiers

2.2 ISL tree-to-tree mappings

With our syntactic representations and the notion of tree tiers firmly in place, it only remains for us to define deterministic *input strictly local* (ISL) tree-to-tree transductions before we start our investigation of movement as a subregular transduction in §3.

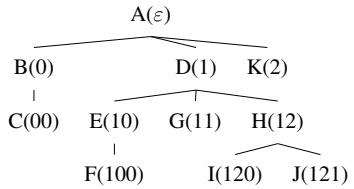
Deterministic ISL transductions, also called *ISL mappings*, were first defined in subregular phonology for the string-to-string case (Chandlee, 2014, 2017; Chandlee and Heinz, 2018). The ISL string-to-string mappings were subsequently generalized to (non-deterministic) tree-to-tree transductions in Graf (2020). An ISL tree transduction τ is specified by a finite number of rewrite rules. The left-hand side consists of a tree with one distinguished node h that is to be rewritten — the rest of the tree just provides the strictly local context in which this specific rule must be applied to h. The right-hand side consists of a tree with indexed *ports* $\square_1$, $\square_2$, …, $\square_n$ ($n \geq 0$) such that each $\square_i$ is filled with the output of τ for the i-th daughter of h. Figure 2 gives a simple example for mapping a dependency tree without movement (and with at most two arguments per LI) to its corresponding bare phrase structure tree — the reader is advised to study this example carefully before moving on to the formal definition.

We first put in place some common concepts from the tree transducer literature. A Σ-tree is a finite tree over alphabet Σ. We assume that all Σ-trees have a finitely bounded branching factor. Given a Σ-tree t, each node n in t is given a unique Gorn address $a(n)$ (Gorn, 1967): $a(n) = \varepsilon$ if n is the root of t, and otherwise $a(n) = ui$, where u is the Gorn address of the mother of n and i is the number of left siblings of n. A Σ-*tree context* c is the result of replacing $n \geq 1$ leaves in a Σ-tree with distinguished symbols drawn from a set of ports, which are denoted with $\square_i$, $i \geq 1$. Given such a context c and Σ-trees or Σ-tree contexts $t_1, \ldots, t_n$, $c\{1 : t_1, \ldots, n : t_n\}$ is the result of replacing $\square_i$ in c with t_i.

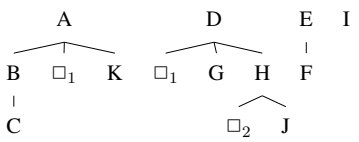
In order to determine the configurations in which ISL rewrite rules may apply, we introduce the notion of a tree disassembly.

Definition 4 (Tree dissassembly). A *disassembly* of tree t at addresses b, ba_1, …, ba_n is an $(n+2)$-tuple that consists of I) t with the subtree s at b replaced with $\square_1$, II) s with the subtrees at addresses ba_1, …, ba_n replaced with $\square_1$, …, $\square_n$, and III) the subtrees at addresses ba_1, …, ba_n. ⌟

Example. Consider the tree t below, with each node followed by its Gorn address in parentheses.



The disassembly of t at addresses 1, 10, and 120 consists of the following trees/contexts:



Next we define what ISL rewrite rules may look like and how a given rule may apply within a tree.

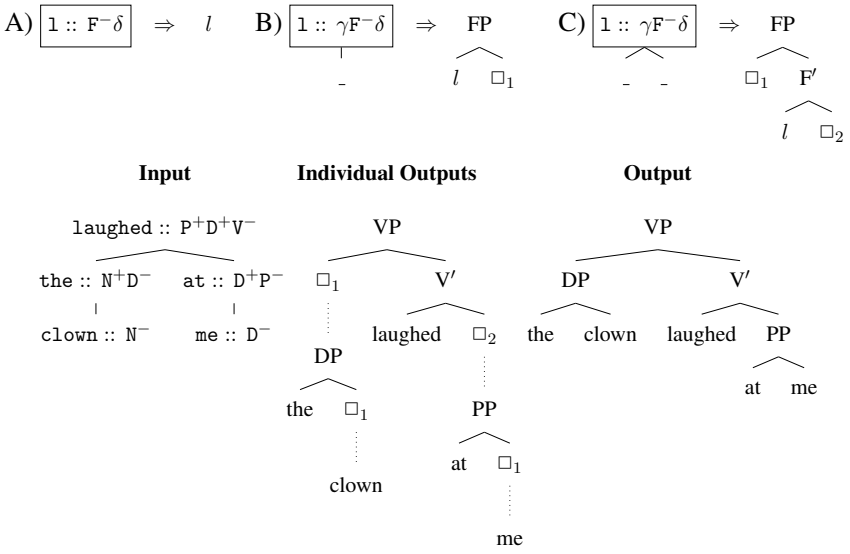
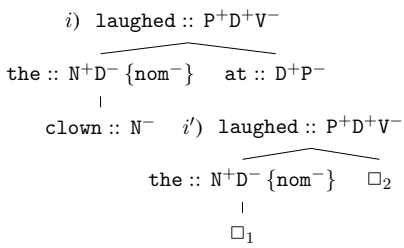


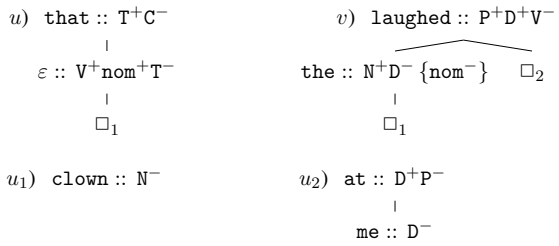
Figure 2: ISL rewrite rules for converting movement-free dependency trees to bare phrase structure trees (top) with example (bottom); boxes around nodes indicate which nodes should be rewritten, γ is a non-empty string of selector features, δ is a (possibly empty) set of licensee features, and _ matches any node

Definition 5 (ISL rewrite rule). An *ISL rewrite rule* is a triple $r := \langle i, a, o \rangle$ where the *input environment* i is a Σ-tree, a is the Gorn address of some node in i, and the *output context* o is a Σ-tree context. Suppose w.l.o.g. that i has exactly n leaf nodes at addresses $a_1, \ldots, a_n$ ($n \geq 1$) and let i' be the result of replacing each node at address a_j with $\Box_j$ ($1 \leq j \leq n$). Then r *matches* tree t at address b iff t has a disassembly $\langle u, i', u_1, \ldots u_n \rangle$ at addresses $b, ba_1, \ldots, ba_n$ such that both of the following hold: I) $t = u\{1 : i'\{1 : u_1, \ldots, n : u_n\}\}$, and II) for $1 \leq j \leq n$, the node at address a_j in i has the same label as the node at address ba_j in t. A node at address ba in t can be rewritten by r iff r matches t at address b. ⌟

Example. Consider a rewrite rule $r := \langle i, a, o \rangle$, with $a = 0$ and i as shown below (together with its counterpart i'):



Note that the ports of i' have addresses $a_1 := 00$ and $a_2 := 1$. We show that i matches the dependency tree t in Fig. 1 at address $b = 00100$. We first disassemble t at addresses b, $ba_1 = 0010000$, and $ba_2 = 001001$. This yields four trees/contexts u, v, u_1, u_2. For space reason, we only show the subtree of u rooted in that :: T⁺C⁻.



As v is identical to i', it holds that $t = u\{1 : v\{1 : u_1, \ldots, n : u_n\}\} = u\{1 : i'\{1 : u_1, \ldots, n : u_n\}\}$. It is also the case that the nodes of i at addresses a_1 and a_2 have the same labels as the nodes in t at addresses $ba_1 = 0010000$ and $ba_2 = 001001$. Taken together, this means that r matches t at address b. Consequently, r can rewrite as o the node at address $ba = 001000$ in t, which is the :: N⁺D⁻. Note that if the root of i had a third daughter labeled, say, *maliciously*, i would no longer match t at any address.

Definition 6 (Deterministic ISL transduction). Given a set R of ISL rewrite rules, we say that R is *deterministic* iff there are no two rewrite rules $\langle i_1, a_1, o_1 \rangle$ and $\langle i_2, a_2, o_2 \rangle$ in R such that $o_1 \neq o_2$ and there exists a Σ-tree t and node n of t such that n can be rewritten by both rewrite rules.

For each deterministic set R of ISL rewrite rules,

$R(t, n)$ denotes the unique output context o for node n in tree t (if no such o exists, $R(t, n)$ is undefined). We extend this to t in a recursive fashion: if t contains only node n, then $R(t) := R(t, n)$, and if $t := n(s_1, \ldots, s_z)$ (each s_i a Σ-tree), then $R(t) := R(t, m)\{1 : R(t, d_1), \ldots, z : R(t, d_z)\}$. A tree-to-tree transduction τ with domain D is *deterministic input strictly local* iff there is a finite deterministic set R of ISL rewrite rules such that $\tau(t) = R(t)$ for all $t \in D$. In this case, we also call τ an *ISL (tree-to-tree) mapping*. ⌟

3 Movement as a subregular transduction

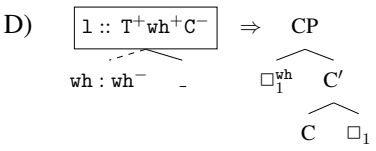
Move cannot be captured with ISL tree-to-tree mappings. The problem is not with the determinism of those mappings. In the formalism used in this paper, Move is a deterministic operation in the sense that the landing sites of a mover can be inferred deterministically from LIs' feature annotations (and as a result the definition of ISL mappings in this paper can safely avoid many complexities in the definitions of non-deterministic ISL transductions in Graf 2020). But while movement is deterministic, it is also unbounded — a mover and its target site can be arbitrarily far apart. Since ISL transductions must be definable in terms of a finite set of rewrite rules, and since each rewrite rule $\langle i, a, o\rangle$ is limited to the finite structural context given by i, ISL transductions cannot handle such unbounded dependencies. For example, we may want to rewrite a node n that carries $\texttt{wh}^+$ as a phrase whose specifier is filled by a wh-mover, but our rewrite rules provide no means to refer to this mover unless it happens to be very close to n. In order to capture movement, ISL rewrite rules must be able to refer to nodes that can be arbitrarily far away.

Tiers provide a natural solution to this problem. We already saw in §2.1 that tiers play a key role in movement — even though movement is unbounded over dependency trees, it is local over tiers. All we have to do is to incorporate this tier-based locality into ISL transductions.

Suppose, then, that we enrich our rewrite rules with another type of ports, called *tier ports*. If we are to rewrite a node n that is part of some `f`-tier, then its output context can include `f`-tier ports. The left-hand side of rewrite rules now also specify a specific test, and a tier port can only pick out the node that passes this test (the node must be unique!). The use of tier tests in the rewrite rules is why I call this new class of transductions *ISL tree-to-tree mappings with TSL tests*.

In this paper, the TSL tests are particularly simple as each one corresponds to a fixed set of LIs that pass the test. Just like the licensing function of TSL in §2.1 could define string languages of various complexity levels all the way up to recursively enumerable, the tests for tier ports can be of arbitrary complexity. But at least for movement, the maximally restricted class of lexical tests (in subregular parlance, SL-1 tests) is sufficient. Hence this paper restricts itself to the even weaker subclass *ISL tree-to-tree mappings with lexical TSL tests*.

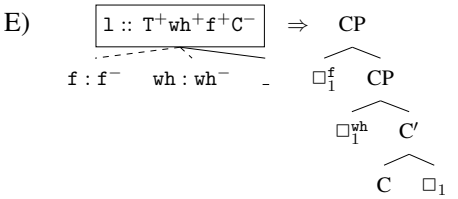
Let us consider how this system captures simple cases of movement. To this end, we add a new rewrite rule to the set in Fig. 2.



This rule targets C-heads that select a TP and provide a landing site for wh-movement. Every such C-head is rewritten as a CP where the complement is filled by the output of the first daughter in the dependency tree, whereas the specifier is filled by the output of the unique node x such that the C-head is the `wh`-tier mother of x and x carries $\texttt{wh}^-$. This is sufficient to connect movers to their landing sites.

Rule D uses two new notational devices: dashed lines for the tier mother-of relation, and tier ports. The dashed line in D leads to a special node that starts with the name of a tier (`wh` in this case), followed by a colon, and the set of LIs on this tier that can be picked out by the tier port $\square_1^{\texttt{wh}}$. Here $\texttt{wh}^-$ is used as a shorthand for the set of all LIs that carry $\texttt{wh}^-$. The tier port $\square_1^{\texttt{wh}}$ is to be filled with the output of the unique node that is a `wh`-tier daughter of the node to be rewritten and carries $\texttt{wh}^-$.

In a more elaborate case where the C-head also attracts some other kind of `f`-mover, the rule would look as follows.



A fully worked out example is shown in Fig. 3 for the sentence *who said that*, where the subject *who*

first undergoes subject movement to Spec,TP and then wh-moves to Spec,CP.

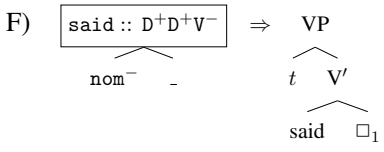
Quite generally, adding lexical TSL tests to ISL tree-to-tree mappings only requires three minor tweaks. First, each rewrite rule is extended to also include a finite (and possibly empty) collection of TSL tests. Second, the notion of a rewrite rule *matching* a tree at a given address b is expanded to also require partial tier matches: if the rule specifies that the node at address a is an $\texttt{f}$-tier mother of an element that passes some test ϕ, then the node at address ba in the dependency tree must be part of the $\texttt{f}$-tier and must have exactly one node x among its $\texttt{f}$-tier daughters such that x passes test ϕ. Finally, the definition of $R(t)$ is amended to include substitution into tier ports. The full definition that incorporates all these changes is given in the appendix.

Inspection of the example in Fig. 3 quickly reveals that the solution laid out above does not quite work as expected for movement. It attaches every mover to all its landing sites, and as a result the bare phrase structure tree contains multiple instances of *who*. In other words, the rewrite rules above implement a copy-theory of movement, but they do not capture the fact that moved phrases are only pronounced in their final landing site. A solution is readily available, though, provided one can tell the final movement step of a mover just from its feature make-up.

4 Linearization and the output-oriented BoIM

Our previous solution for movement runs into problems because movement actually consists of two steps: attaching the mover to all its landing sites, and delinking it from all positions that are not its final landing site.

Delinking itself is fairly simple from the perspective of ISL transductions. Consider the example below for delinking the moving *who* in Fig. 3 from its base position under *said*.



Here $\texttt{nom}^-$ is a shorthand for any LI carrying $\texttt{nom}^-$. The rewrite rule thus replaces the left daughter with a trace provided it undergoes subject movement. Note that since we only care about well-formed dependency trees where every licensee feature has a matching licensor feature on some other node, the fact that the left daughter carries $\texttt{nom}^-$ guarantees that it will undergo subject movement and hence should not be linearized as an argument of the verb. The feature make-up of the LI thus determines whether its base position should be replaced with a trace.

Things are trickier, though, when we consider intermediate landing sites such as Spec,TP for *who*. Since licensee features are not ordered, we cannot tell whether $\texttt{who} :: \texttt{D}^- \{\texttt{nom}^-, \texttt{wh}^-\}$ first undergoes nom-movement or wh-movement. The assumption that licensee features are unordered is crucial for the tier-based perspective of movement, it cannot be easily done away with. It seems, then, that our delinking trick for base positions does not carry over to intermediate landing sites like Spec,TP. We cannot tell from the local context of the T-head whether the subject mover with $\texttt{nom}^-$ will move on to a higher position via wh-movement, or if it has already done so and will thus stop in Spec,TP. One may be tempted to try ideas like merging the nom-tier and the wh-tier into a single tier, but these do not work either because then a mover and its landing site may no longer stand in a mother-daughter configuration. While a mathematical proof is still outstanding, it seems that there is no way in the current system to correctly distinguish final from intermediate landing sites.

Linguists will point out, though, that Spec,TP cannot be the final landing site for *who* due to the *Ban on Improper Movement* (BoIM): once a mover undergoes an instance of A′-movement like wh-movement, it can no longer undergo any A-movement steps such as subject movement. The BoIM rules out sentences like the illicit *who wonders [t John saw t]*, where *who* first wh-moves to Spec,CP of the embedded clause before undergoing subject movement into the matrix clause.

In light of the BoIM, it is readily apparent from the feature make-up of $\texttt{who} :: \texttt{D}^- \{\texttt{nom}^-, \texttt{wh}^-\}$ that it first undergoes nom-movement and then wh-movement. Consequently, the purely feature conditioned delinking strategy still works and one could something like rule G below for rewriting the T-head. Rule H for rewriting the C-head looks almost exactly the same except that we insert the mover and not a trace. In both rules, $\{\texttt{nom}^-, \texttt{wh}^-\}$ matches every LI that carries at least those two licensee features.

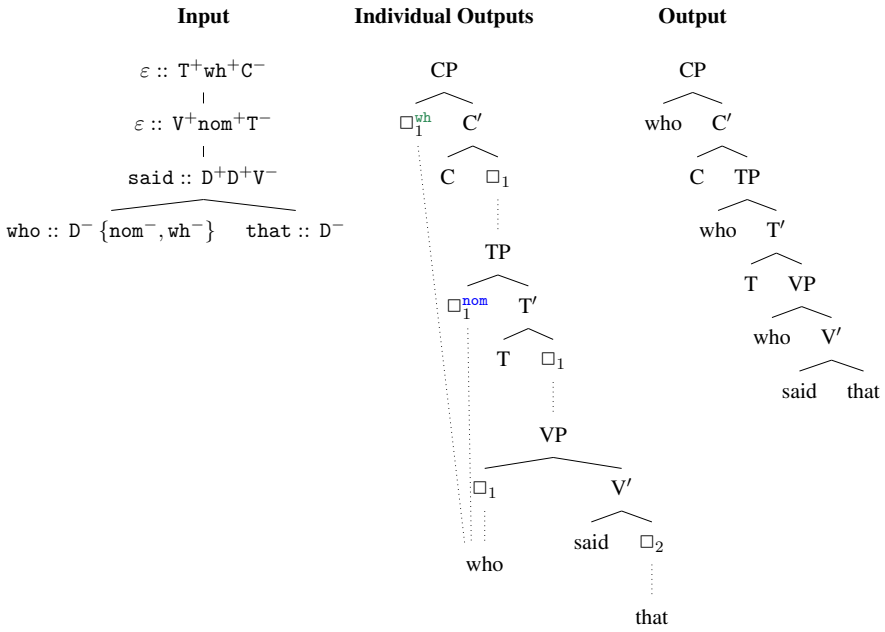
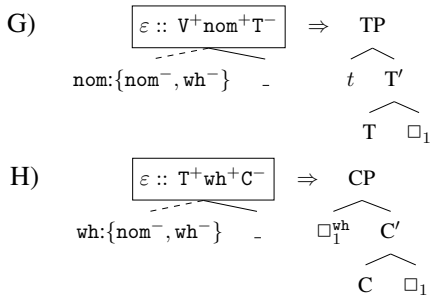


Figure 3: The dependency tree for *who said that* is rewritten into the corresponding bare phrase structure tree.



At least in the case of subject movement and wh-movement, then, ISL tree-to-tree transductions with TSL tests allow us not only to associate a mover with all its landing sites, but also to produce linearized output structures with the correct string yield.

In order for this solution to extend to all of syntax, however, a stronger property has to be in place.

Definition 7 (Output-oriented BoIM). For no LI l with set $\{\mathtt{f}_1^-, \ldots, \mathtt{f}_n^-\}$ of licensee features may there be well-formed dependency trees t_1 and t_2 such that I) both t_1 and t_2 contain l, and II) l's final movement step is $\mathtt{f}_i$ in t_1 and $\mathtt{f}_j$ in t_2 ($i \neq j$). ⌟

In other words, for every LI l one can always predict its final movement step based purely on inspection of the LI itself.

I conjecture that the output-oriented BoIM is a universal property of movement across languages. This is prompted by two observations. First, a preliminary analysis of the MG corpus (Torr, 2017) suggests that the output-oriented BoIM holds for the trees in that treebank. In fact, the licensee features used in that corpus seem to obey an even stronger restriction: for every LI l that carries, say, $\mathtt{f}^-$ and $\mathtt{g}^-$, it is always the case that l undergoes $\mathtt{f}$-movement before $\mathtt{g}$-movement. While corpora represent just a finite slice of a possibly infinite range of licit configurations, it is encouraging that the conjecture clears this first hurdle with ease.

The second argument is more indirect: While the syntactic literature has noted potential exceptions to the BoIM, those do not directly carry over to the system used here. Consider the case of hyperraising in Zulu (see Zyman 2023 and references therein). Here a DP undergoes A-movement from a position in the embedded clause to some position in the matrix clause, yielding a configuration similar to the illicit English sentence *Mary seems [that will go home]*. Minimalists assume for independent reasons that *Mary*, rather than moving directly from the embedded subject position to the matrix subject position, has to stop in Spec,CP of the embedded clause. As the latter is an instance of A′-movement, hyperraising seems to involve an A′-movement step to Spec,CP followed by A-movement to the subject position of the matrix clause. But this A′-movement step is driven by theoretical considerations related to successive cyclic movement, which is treated very differently

in MGs and subregular syntax. The phenomena that are used to motivate successive cyclic movement, e.g. wh-agreement in Irish, can be captured without such movement in TSL syntax (Graf, 2022c). Without successive cyclic movement, though, hyperraising is no longer a counterexample to the standard BoIM, let alone the output-oriented BoIM that is needed in this system of ISL transductions with lexical TSL tests.

If the output-oriented BoIM turns out to be empirically robust, then the limits of ISL tree-to-tree transductions with TLS tests provide a novel motivation for the otherwise mysterious BoIM (which would then be a stronger implementation of the output-oriented BoIM). Subregular complexity might offer a computational third-factor explanation (Chomsky, 2005) for one of the most robust universals of syntax.

5 Remarks and open issues

The discussion so far has assumed that all movement steps are overt. Minimalist syntax and MGs both allow for covert movement steps, which do not affect linearization. In such systems, the final landing site of LI l with respect to linearization may be distinct from the landing site of its final movement step. This does not introduce any new challenges, though, as long as the following condition is met: for every set $S := \{\mathtt{f}_1^-, \ldots, \mathtt{f}_n^-\}$ of licensee features and every type of output structure (e.g. phrase structure tree, LF), one can tell directly from S whether $\mathtt{f_i}$-movement ($1 \leq i \leq n$) creates a copy or a trace at the landing site.

Another issue arises with successive cyclic movement. A common approach in MGs posits that successive cyclic movement is not feature-triggered but rather a result of the output mapping inserting traces and/or copies at specific positions along a movement path. ISL mappings with lexical TSL tests struggle with this because a node that is not on tier T cannot use T to test whether it is along a movement path. At the same time, putting, say, all C-heads on a tier T together with all wh-movers does not help either as the T-daughter of some C-head may then just be another C-head rather than the desired wh-mover. Instead of a transduction-based model of successive cyclic movement, one based on tier constraints may be more promising (cf. Graf, 2022c).

Finally, the complexity of copies vs. traces merits further exploration. Kracht (2001) observes that one can freely translate between copies and traces, but we saw that copy-based movement is simpler than trace-based movement because the latter requires additional restrictions on movement. Similarly, transductions with copying are more complex than linear transductions, yet the latter are sufficient for trace-based movement. This suggests that the subregular notions of complexity crosscut traditional ones in unexpected ways that may sometimes favor more complex machinery in one area in order to reduce complexity in another. These connections could only be hinted at in this paper but are ripe for future exploration from a mathematical perspective, e.g. in terms of DAG transductions (Drewes, 2017) as dependency trees with tier relations are essentially DAGs with labeled edges.

Conclusion

I have introduced (deterministic) ISL tree-to-tree transductions with TSL tests as a new class of subregular transductions that expands the ISL tree-to-tree transductions of Graf (2020) with the tier-based view of movement in Graf (2018, 2022c) in order to provide a subregular model of movement as a mapping from syntactic derivations (represented via dependency trees) to output structures. This class of transductions is still conceptually simple while offering enough expressivity to easily relate each mover to all its landing sites. The transductions in this class are too weak to distinguish final from intermediate landing sites, which is essential for obtaining the correct string yield from a syntactic derivation. However, it seems that a variant of the Ban on Improper Movement restricts syntax in just the right way to draw the necessary distinction between final and intermediate landing sites based purely on the feature make-up of the mover. It remains to be seen whether the output-oriented BoIM proposed here is indeed empirically viable, but the possibility is tantalizing as it promises a computational grounding for one of the best-known and most robust syntactic constraints.

Acknowledgments

This paper is dedicated to Christopher Graf, who entered this world a bit ahead of schedule, on the day of the submission deadline. I thank the reviewers for pushing this paper in a linguistically more comprehensive direction. The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344.

References

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee. 2017. Computational locality in morphological maps. *Morphology*, 27:599–641.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry*, 49:23–60.

Noam Chomsky. 2005. Three factors in language design. *Linguistic Inquiry*, 36(1):1–22.

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *Formal Grammar*, pages 35–50, Berlin, Heidelberg. Springer.

Frank Drewes. 2017. On DAG languages and DAG transducers. *Bulletin of the European Association for Theoretical Computer Science*, 121.

Saul Gorn. 1967. Explicit definitions and linguistic dominoes. In *Systems and Computer Science, Proceedings of the Conference held at University of Western Ontario, 1965*, Toronto. University of Toronto Press.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, pages 117–136.

Thomas Graf. 2020. Curbing feature coding: Strictly local feature assignment. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*, pages 362–371.

Thomas Graf. 2022a. Diving deeper into subregular syntax. *Theoretical Linguistics*, 48:245–278.

Thomas Graf. 2022b. Subregular linguistics: Bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48:145–184.

Thomas Graf. 2022c. Typological implications of tier-based strictly local movement. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2022*, pages 184–193.

Thomas Graf, Alëna Aksënova, and Aniello De Santo. 2016. A single movement normal form for Minimalist grammars. In *Formal Grammar: 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016*, pages 200–215, Berlin, Heidelberg. Springer.

Thomas Graf and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 12–26, Toronto, Canada. Association for Computational Linguistics.

Thomas Graf and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist grammars. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*, pages 275–285.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of SIGMORPHON 2018*, pages 151–160.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frank Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. Mouton De Gruyter.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Shiori Ikawa, Akane Ohtaka, and Adam Jardine. 2020. Quantifier-free tree transductions. In *Proceedings of the Society for Computation in Linguistics (SCiL)*, volume 3, pages 455–458.

Adam Jardine. 2016. Computationally, tone is different. *Phonology*, 33:247–283.

Jing Ji and Jeffrey Heinz. 2020. Input strictly local tree transducers. In *Language and Automata Theory and Applications: 14th International Conference, LATA 2020, Milan, Italy*, volume 12038 of *LNCS*, pages 369–381.

Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to Minimalism. In *Model Theoretic Syntax at 10*, pages 71–80.

Marcus Kracht. 2001. Syntax in chains. *Linguistics and Philosophy*, 24:467–529.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from Uyghur backness harmony. In *Proceedings of Formal Grammar 2018*, pages 62–83, Berlin. Springer.

Kevin McMullin. 2016. *Tier-Based Locality in Long-Distance Phonotactics: Learnability and Typology*. Ph.D. thesis, University of British Columbia.

Luigi Rizzi. 1990. *Relativized Minimality*. MIT Press, Cambridge, MA.

Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.

Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.

John Torr. 2017. Autobank: a semi-automatic annotation tool for developing deep Minimalist grammar treebanks. In *Proceedings of the Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 81–86.

Erik Zyman. 2023. Raising out of finite clauses (hyperraising). *Annual Review of Linguistics*, 9:29–48.

Definition of ISL mappings with lexical TSL tests

We now allow tree contexts to also contain *tier ports*, which are ports that are indexed with the name of a tier, e.g. $\square_i^T$. We also amend our tree substitution notation to allow for the use of tier ports: $c\{Ti : t\}$ is the result of replacing tier port $\square_i^T$ in context c with t. The indices of tree ports will be interpreted slightly differently from standard ports. Whereas $\square_i$ refers to the (output of the) i-th daughter of the node being rewritten, $\square_i^T$ will refer to the (output of the) node picked out by the i-th TSL test over tier T.

A *lexical TSL test over tier* T is a formula of the form $\phi_T(n, x) := n \lhd_T x \wedge x \in U$, where U is some subset of T. To avoid various complications related to non-determinism, we only consider the special case where $\phi_T(n, x)$ is *deterministic* over some set L of trees. That is to say, for every $t \in L$ and node n of t, there is at most one x such that $\phi_T(n, x)$ is true. We also call $\phi_T(n, x)$ *L-deterministic*. Slightly abusing notation, we let $\phi_T(t, n)$ denote the unique node x (if it exists) such that $\phi_T(n, x)$ holds in t. Finally, we define Φ as a finite family of lexical TSL tests $\phi_{T_1,1}, \ldots, \phi_{T_1,z_1}, \ldots, \phi_{T_k,1}, \ldots, \phi_{T_k,z_k}$ indexed by pairs of tier names and positive natural numbers.

An *ISL rewrite rule with lexical TSL tests over tiers* T_1, …, T_k is a pair $\langle r, \Phi\rangle$ such that $r := \langle i, a, o\rangle$ is an ISL rewrite rule (where o may contain tier ports). We say that $\langle r, \Phi\rangle$ is L-deterministic iff every $\phi_{T,i} \in \Phi$ is L-deterministic. Given such an L-deterministic rule $\rho := \langle r, \Phi\rangle$ and tree $t \in L$, ρ *matches* t at node n with address b iff I) r matches t at address b, and II) for every $\phi_{T,i} \in \Phi$, $\phi_{T,i}(t, n)$ exists. As with ISL rewrite rules, a node at address ba in t can be rewritten by $\rho := \langle\langle i, a, o\rangle, \Phi\rangle$ iff ρ matches t at address b.

A set R of ISL rewrite rules with TSL tests over tiers T_1, …, T_k is *L-deterministic* iff $\{r \mid \langle r, \Phi\rangle \in R\}$ is a deterministic set of ISL rewrite rules and every $r \in R$ is L-deterministic. Note that this excludes any set R containing at least two rules that only differ in their TSL tests.

Given such an L-deterministic set R, $R(t, n)$ denotes the unique output context o for node n in tree $t \in L$. We extend this to t in a recursive fashion: If t contains only node n, then $R(t) := R(t, n)$. If $t := m(d_1, \ldots, d_z)$, then $R(t)$ is

$$\begin{aligned} R(t, m)\{&1 : R(t, d_1), \ldots, z : R(t, d_z),\\ &T_1 1 : R(t, \phi_{T_1,1}(t, m)), \ldots,\\ &T_1 z_1 : R(t, \phi_{T_1,z_1}(t, m)), \ldots,\\ &T_k 1 : R(t, \phi_{T_k,1}(t, m)), \ldots,\\ &T_k z_k : R(t, \phi_{T_k,z_k}(t, m))\} \end{aligned}$$

A tree-to-tree transduction τ with domain D is *deterministic input strictly local with lexical TSL tests* iff there is a finite set R of ISL rewrite rules with TSL tests such that R is deterministic over D and $\tau(t) = R(t)$ for all $t \in D$. In this case, we also call τ an *ISL (tree-to-tree) mapping with lexical TSL tests*.

Text segmentation similarity revisited: A flexible distance-based approach for multiple boundary types

Ryan Ka Yau Lai, Yujie Li
University of California, Santa Barbara
{kayaulai, yujie_li}@ucsb.edu

Shujie Zhang
University of California, Berkeley
z4362687@berkeley.edu

Abstract

Segmentation of texts into discourse and prosodic units is a ubiquitous problem in corpus linguistics and psycholinguistics, yet best practices for its evaluation – whether evaluating consistency between human segmenters or humanlikeness of machine segmenters – remain understudied. Building on segmentation edit distance (Fournier & Inkpen 2012, Fournier 2013), this paper introduces a new measure for evaluating similarity between two segmentations of the same text with multiple, mutually exclusive boundary types, accounting for varying identifiability and confusability between these types. We implement a dynamic programming algorithm for calculation specifically geared towards this type of segmentation problem, apply it to a case study of intonation unit segmentation measuring inter-annotator agreement, and make suggestions for interpreting results.

1 Introduction

In computational corpus linguistics and psycholinguistics, many types of annotation and experimental tasks can be seen as *segmentation* problems, where a text is broken up into segments. These segments can be morphemes, tokens (i.e. tokenisation), prosodic, syntactic and interactional units (such as intonation units, sentences, utterances and turns), as well as larger segments of discourse like topics.

When multiple annotators, whether human or machine, have annotated the same text, the question arises as to how to measure the degree of divergence. There are multiple motivations for this question. Methodologically, we often want to evaluate annotation schemes and annotator training (e.g. Lin 2009), as well as humanlikeness of computational segmentation models. Theoretically, comparing the consistency of different types of segmentation sheds light on human perception of boundaries, such as how boundaries are perceived (e.g. Troiani et al. 2023).

This paper focuses on one type of problem: segmentation using a set of mutually exclusive boundary types. Punctuation prediction (Lu & Ng 2010), for example, can be seen as this task: a text is divided using a set of mutually exclusive punctuation marks. Consider, as an example, the following unpunctuated, 5-word text:

London Bridge is falling down

We assume that each word is potentially followed by a punctuation mark, so there are 5 possible spots to place boundaries. Under this situation, there cannot be more than one punctuation mark between two words (unlike an application where, for example, one segments a text into both sentences and paragraphs, and hence a space may be both a sentence boundary and a paragraph boundary). Assuming there are three candidate punctuations, comma, period and question mark, the sets of choices are thus (Ø represents no boundary):

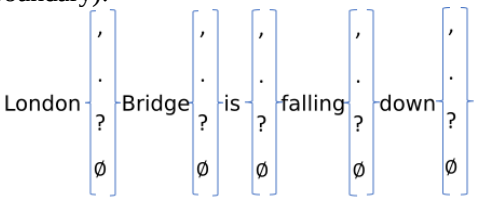


Figure 1: Schematic illustration of the type of segmentation problem explored in this paper, where each potential boundary can be one of a fixed set of mutually exclusive boundary types.

In the computational literature, various metrics for evaluating segmentation differences have been proposed and examined (e.g. Beeferman et al.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 300-309.
Amherst, Massachusetts, June 15-17, 2023

1999, Pevzner & Hearst 2002, Lamprier et al. 2007, Franz et al. 2007, Peshkov et al. 2013, Peshkov & Prévot 2014). To our knowledge, however, none are specifically geared towards this type of problem. An additional complication of this paper is that our method must work for both monologic and dialogic texts, which none of the previous methods have focused on.

In this paper, building on Fournier & Inkpen (2012) and Fournier (2013), we propose a new metric, *flexible segmentation similarity* (S_f), allowing not just for gradient similarities between boundary types, as discussed also by Fournier (2013), but also for differentiating insertion and deletion of different boundary types. We also discuss a simulation-based approach to calculate Cohen's κ inter-annotator agreement for this measure. We apply the method to a case study of intonation unit (IU) segmentation, where part of the NCCU Taiwan Mandarin Corpus (Chui & Lai 2008) was manually segmented into IUs, and each IU boundary was classified according to boundary intonation preceding it.

2 Previous work

Many evaluation metrics have been applied to segmentation, including match percentage (Lin 2009), conventional measures of classification performance like precision, recall, F1 value and accuracy, windows-based approaches like P_k (Beeferman et al. 1999) and WindowDiff (Pevzner & Hearst 2002), and edit distance-based methods (Fournier & Inkpen 2012, Fournier 2013). Our measure builds on the last approach due to significant disadvantages of the rest.

2.1 Problems with non-edit distance-based metrics

The pros and cons of these methods are widely discussed in the literature (e.g. Beeferman et al. 1999, Pevzner & Hearst 2002, Lamprier et al. 2007, Franz et al. 2007, Fournier & Inkpen 2012, Peshkov et al. 2013, Peshkov & Prévot 2014), but several problems stand out. Firstly, conventional classification performance measures and match percentage fail to account for 'near-misses' (Pevzner & Hearst 2002), where two annotators place the 'same' boundary in different but close locations. This is often the case in intonation unit boundary identification: different boundary-marking acoustic cues can be spread across multiple words, creating fuzzy boundaries (Barth-Weingarten 2016: 6-7). Treating the problem as simple classification unduly penalises such cases. For example, Troiani et al. (2023) find that English speakers have very poor performance on segmenting Kazakh texts into intonational boundaries when the many near-misses are ignored; this was likely due to uncertainty about which part of the recording corresponded to which part of the transcript.

Secondly, conventional classification performance measures and windows-based metrics are asymmetric (Fournier 2013): We evaluate one annotation set against another; switching the places of the two annotations results in different numbers. So these measures can only compare one annotation against a gold standard, but not when there is no ground truth (e.g. between two equally-trained human annotators).

Thirdly, all non-edit-distance-based measures do not account for multiple boundary types, which often arise in corpus linguistics, and treat all mistakes as 'equal', ignoring differences in difficulty between boundary types (cf. Qian et al 2016 in the tokenisation context).

2.2 Edit distance-based metrics

The edit distance-based approaches Segmentation Similarity (*S*) (Fournier & Inkpen 2012; henceforth F&I) and Boundary Similarity (*B*) (Fournier 2013) are the closest to our proposed measure, as they account for near-misses, are symmetric, and allow multiple boundary types. They are briefly reviewed here with our simplified notation, which will be used throughout this paper.

In the following, we will refer to the elements between which boundaries can be added as *tokens*. This may be roughly words in tasks like intonation unit segmentation, or a larger unit like turn-constructional units in turn segmentation, sentences in topic segmentation, and so on. For *S* and *B*, the number of potential boundaries *N* is the number of tokens minus 1. The potential boundaries in a text will be denoted $b_1, b_2, \ldots, b_{N-1}$; for example, in the Figure 1 example, b_1 is between *London* and *Bridge*, b_2 between *Bridge* and *is*, and so on. The actual boundaries from annotator *i* will be denoted $b_{i,1}, b_{i,2}, \ldots, b_{i,N-1}$. Since these measures deal with non-mutually exclusive boundary types, each of $b_{i,1}, b_{i,2}, \ldots, b_{i,N-1}$ is a *set* of boundaries. For example, when simultaneously annotating turn

and sentence boundaries, one potential boundary could be both a turn boundary and a sentence boundary.

For calculating *S*, one set of annotations is transformed into another, minimising the number of operations taken. There are three possible operations: a) adding boundaries, b) deleting boundaries, and c) transposing boundaries, i.e. moving a boundary to a different position, in order to align it with a boundary placed by another annotator. Thus, if one annotator put a boundary in $b_{1,1}$ but not in $b_{1,2}$, and another put a boundary of the same type in $b_{2,2}$ but not in $b_{2,1}$, then we can transpose the boundary from $b_{1,1}$ to $b_{1,2}$ to match the second annotator. This only takes one operation, as opposed to deleting in $b_{1,1}$ and adding it to $b_{1,2}$, which takes two, thus preventing the problem of overpenalising near-misses.

The similarity is then calculated thus:

$$S = \frac{N - \#(\text{edits})}{N} = 1 - \frac{\#(\text{edits})}{N}$$

S is thus a ratio in [0, 1]: the larger *S* is, the closer the annotations. F&I also mention the possibility of scaling the number of boundaries 'moved' in transposition so that e.g. 2 transpositions might count for fewer than two edits.

B differs from *S* in two ways. Firstly, the normalisation is different. The score is normalised by the number of edits plus the number of correct boundaries. This in essence means the number of total boundaries perceived by the two annotators, assuming that transposed boundaries are the 'same' boundary across annotators. This prevents biasing annotators towards a smaller number of boundaries, i.e. longer segments. This is useful for tasks like intonation unit segmentation: in languages like English and Mandarin, intonation unit boundaries in spoken language are typically denser than punctuation boundaries in written language. Annotators may be influenced more by orthography if biased towards fewer boundaries.

Secondly, instead of the number of edits, the distance between the two annotations is calculated more flexibly by assigning different costs to different edit operations. Although addition and deletion retain the cost of 1, *B* allows for substitutions between boundary types. For *B*, boundary types are organised on an ordinal scale, and the cost of substituting one boundary for another is their distance on their ordinal scale normalised by the total number of boundary types. The formula for *B* is as follows:

$$B = 1 - \frac{C_{total}}{\#(\text{edits}) + \#(\text{correct boundaries})}$$

where C_{total} is the total cost of operations.

Although *S* and *B* are excellent measures of similarity between different annotations, they still have disadvantages. Firstly, although *B* allows for different similarity between different boundary types, recognising that some boundaries may be more confusable than others, it makes the strong assumption that these differences are gradable on an ordinal scale, which is problematic for intonation unit segmentation (see Section 4).

Secondly, by setting addition and deletion cost by default to 1, it ignores the fact that some boundaries may be easier to identify than others. Deleting an easy boundary should cost more than deleting a difficult one.

Finally, *S* and *B* are excellent for written and monologic texts, but are unsuited for multi-party conversations where tokens are not organised in a single linear sequence, since two people's speech can overlap. Our proposed method addresses all three of these weaknesses.

3 Proposed method

3.1 Definition of S_f

Like Segmentation Similarity (*S*) and Boundary Similarity (*B*), we evaluate similarity first by transforming one annotation to the other and calculating the cost, normalising this, and then subtracting the similarity from 1 to get a distance. We present two options for normalising: following *S* in using the number of potential boundaries (S_f) and following *B* in using the number of edits plus correct boundaries (S_f^B) (*pace* Fournier (2013), we argue (Section 4) that both normalisation approaches can be useful in different situations):

$$S_f = 1 - \frac{C_{total}}{N}$$

$$S_f^B = 1 - \frac{C_{total}}{\#(\text{edits}) + \#(\text{correct boundaries})}$$

The calculation of C_{total} departs substantially from *S* and *B*. We allow for user-defined addition, deletion and substitution costs using the similarity matrix M_T. The values in the matrix are the similarity (on the interval [0, 1]) between the two different boundary types. One minus the value is the cost of substitution between these two boundary types. The final row and column are for the *lack* of a boundary. Here is a sample matrix with two boundary types $T = \{p, q\}$:

$$M_T = \begin{pmatrix} 1 & s_{pq} & s_{p\emptyset} \\ s_{qp} & 1 & s_{q\emptyset} \\ s_{\emptyset p} & s_{\emptyset q} & 1 \end{pmatrix}$$

Here, s_{ab} is one minus the cost of substituting a for b, and $\emptyset$ refers to the lack of a boundary. Thus, $1 - s_{\emptyset q}$ is the addition cost of q, and $1 - s_{p\emptyset}$ is the deletion cost of p. When a symmetric score is desired, e.g. comparing two human annotators, the matrix must be symmetric as well, i.e. $s_{xy} = s_{yx}\ \forall x, y \in T \cup \{\emptyset\}$. This means substituting *x* for *y* has the same cost as substituting *y* for *x*, and insertion and deletion have identical costs. By default, this matrix is the identity matrix I, i.e. all substitutions, additions and deletions have a cost of 1. An example of user-defined M_T will be given in Section 4; in cases where existing annotations by expert annotators are available, confusion matrices from those raters can be used instead to determine M_T in evaluating similarity between novice annotators' work. In the rest of this paper, addition and deletion will be treated as special cases of substitution involving $\emptyset$.

Transposition cost can also be set flexibly for different boundary types, represented by c^t, a vector with as many entries as there are boundary types. In this paper, we will set transposition cost at half of insertion/deletion cost, i.e., $c^t = \frac{1}{2}(\mathbf{1} - \left[s_{p\emptyset}\ \ s_{q\emptyset}\right]^T)$. A glossary of notation is in Table 1.

S_f , S_f^B	Flexible segmentation distance normalised respectively with *N* and $\#(edits) + \#(correct\ boundaries)$
C_{total}	Total cost of transforming between annotations
$\emptyset$	No boundary
N	Number of potential boundaries
b_i	*i*th potential boundary
$b_{i,j}$	Annotator *j*'s annotation of b_i
T	Set of boundary types
M_T	Similarity matrix for T
s_{pq}	Similarity between *p* and *q*
$t_1[x]$	*x*th element of boundary list t_1
$t_1[-x]$	t_1 without the *x*th element
$t_1[x:y]$	*x*th to *y*th elements of t_1
c^t	Vector of transposition costs
$\mathrm{tr}(t_1, x, y)$	boundary list t_1 with the *x*th and *y*th elements swapped
κ	Cohen's kappa
S_f^{chance}	Chance-level similarity

Table 1: Glossary of notation used in this paper.

3.2 Algorithm for calculating S_f

Our algorithm first separates the text by conversational participants, since tokens from the same participant cannot overlap, and thus can be taken as one whole running text. We calculate the cost for each participant separately, then take the sum. Also, in our use cases, the end of the text also has a boundary type, so number of potential boundaries *N* is equal to the number of tokens.

For each participant, we first identify all the potential boundaries where *both* annotators put a boundary, regardless of whether their types match. Our boundary types are mutually exclusive, so when two people put different boundaries in the same place, they can be safely assumed to have *identified* the 'same' boundary, and just *classified* it differently. We treat these as substitutions and store the total cost of these operations.

We then further split the text into smaller lists of boundaries at those points where both annotators have a boundary. Consider the following situation:

	b_1	b_2	b_3	b_4	b_5	b_6
Annotator 1	p		p			q
Annotator 2	q		p	p		p

We will split the text at the points b_1, b_3 and b_6, leaving two boundary lists: $[b_2]$ and $[b_4\ b_5]$.[1] We discard the boundary lists where both annotators have no boundaries since such lists are necessarily identical. In this case, we discard $[b_2]$, retaining only the single list $[b_4\ b_5]$. For the remaining boundary lists, we trim all the **common** leading and trailing $\emptyset$s in both lists, since it is pointless to move boundaries to those locations; this leaves only $[b_4]$ in the example.

We then calculate the similarity between the two annotators for each of these segments. For each segment, we run a recursive algorithm, called parDist, to find the minimum cost of transforming one annotation to the next. At each step, we first trim any common leading and trailing $\emptyset$s again. We then choose the next step depending on properties of the two boundary lists:

- If the two boundary lists have size 1, then we simply return the substitution cost (which is 0 if they are the same boundary, and >0 otherwise).

[1] We assume that annotators will not place a single boundary of indeterminate location in more than one spot; thus, Annotator 2 is committing to there being two distinct boundaries at b_4 and b_5.

- If the length is >1, we look for positions where *both* boundary lists have a boundary. If one such position exists, we perform a substitution at it, then perform parDist on the remaining contiguous portion(s) of the boundary list. For example, if the lists have five elements and this substitution happens at the fourth element, then we run parDist again on two boundary sub-lists: the first three elements and fifth element. If the substitution on this list happens at the first element, then we run parDist on the segment from the second to fifth item.
- If there are no positions where both boundary lists have a boundary, but both lists have at least one non-$\emptyset$ boundary, then we attempt both transposition and substitution and take the minimum. For transposition, we attempt to move the first non-$\emptyset$ boundary in the second boundary list so that it matches up with an element in the first boundary list, then run parDist on the resultant boundary lists. For substitution, we simply replace the first element of the second boundary list with the first element of the first boundary list, then run parDist on the remaining boundaries. We take the transposition cost if it is smaller, and vice versa.
- Finally, if one of the boundaries consists of all 0s, then we perform substitution until all the differences are eliminated.

A rough presentation of parDist in pseudocode is presented in Algorithm 1, where t_1 and t_2 are the two annotations, $t_1[1]$ refers to the first element of the boundary list, $t_1[-1]$ refers to the boundary list without the first element, $t_1[x{:}\,y]$ refers to the *x*th to *y*th elements of t_1, and $\mathrm{tr}(t_1, x, y)$ refers to t_1 with the *x*th and *y*th elements swapped. Figure 2 illustrates the algorithm with a concrete example.

function parDist(t_1, t_2):

remove all **common** leading and trailing $\emptyset$s from t_1 and t_2
if length(t_1) ≤ 1:
  return $1 - s_{t_1[1],t_2[1]}$
else:
  if $t_1[1] = t_2[1]$:
    return parDist $(t_1[-1],\ t_2[-1])$
  else if($t_1[1] \neq \emptyset$ & $t_2[1] \neq \emptyset$):
    return $1 - s_{t_1[1],t_2[1]}$
      + parDist $(t_1[-1],\ t_2[-1])$
  else if $\exists\ i$ such that $t_1[i] \neq \emptyset$ & $t_2[i] \neq \emptyset$:
    take the smallest i
    return $1 - s_{t_1[i],t_2[i]}$
      + parDist $(t_1[1{:}\,i-1],\ t_2[1{:}\,i-1])$
      + parDist $(t_1[i+1{:}\,\mathrm{length}(t_1)],$
      $t_2[i+1{:}\,\mathrm{length}(t_1)])$
  else if $t_1[1] = \emptyset$ & $\exists\ i$ such that $t_1[i] \neq \emptyset$:
    take the smallest i
    return $\min(1 - s_{t_1[1],t_2[1]}$
      $+ \mathrm{parDist}(t_1[-1],\ t_2[-1]),$
      $c^t[t_2[1]] \cdot (i-1)$
      $+ \mathrm{parDist}(t_1,\ \mathrm{tr}(t_2, 1, i)))$
  else if $t_2[1] = \emptyset$ & $\exists\ i$ such that $t_2[i] \neq \emptyset$:
    take the smallest i
    return $\min(1 - s_{t_1[1],t_2[1]}$
      $+ \mathrm{parDist}(t_1[-1],\ t_2[-1]),$
      $c^t[t_2[i]] \cdot (i-1)$
      $+ \mathrm{parDist}(t_1, \mathrm{tr}(t_2, 1, i)))$
  else:
    return $1 - s_{t_1[1],t_2[1]}$
      + parDist$(t_1[-1],\ t_2[-1])$

Algorithm 1: Pseudocode for parDist

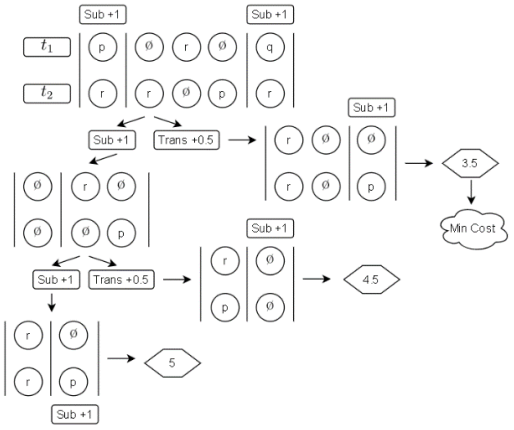


Figure 2: An illustration of parDist, assuming $M_T = I$ (i.e. substitutions including insertion and deletion cost 1), transpositions cost 0.5, and $T = \{p, q, r\}$. Firstly, all positions with a boundary in both annotations are considered substitutions. There are then two options: Either move the *r* of the second annotation to the right, or delete it. In the first case, the *p* must then be deleted, resulting in a cost of 3.5. In the second case, one can then either bring the *p* to the left then substitute it for an *r* (cost = 4.5), or add an *r* and delete the *p* (cost = 5). The minimum cost of all these possibilities is then 3.5.

The actual implementation of the algorithm involves several components omitted from the pseudocode for a cleaner presentation. Two of these components aim at storing information about the process. Firstly, the number of actions hitherto performed is stored and accumulated across iterations of parDist to calculate the

denominator of S_f^B. Secondly, information about each operation – including the operation type, old and new boundary type, and old and new location – can be stored for later access so they can be used for analysing machine segmentation errors or points of inter-annotator disagreement (Section 4 has an example).

Two other components aim at speeding up computation. Firstly, the function stores the minimum cost so far among the total costs that have been calculated. When the cumulative cost in the branch of possibilities currently being explored has exceeded the stored minimum, the function returns NA, thereby aborting the branch, instead of continuing the calculation. Secondly, every time parDist is calculated, the resulting cost and number of operations are stored in a two-dimensional dictionary with t_1 and t_2 (stored as strings) for keys. Before each instance of parDist, the algorithm looks up the dictionary and simply takes the result from there if the operation has been done before. These result in significant speed gains, especially when calculating similarity between simulated annotations for inter-annotator agreement (see Section 3.3).

A property of this algorithm is that a boundary may be both transposed and substituted if the cost of doing so is lower than insertion plus deletion. For the calculation of $\#(edits)$ in the S_f^B formula, such edits will only be counted once, in the spirit of normalising by the total number of boundaries. A consequence of this property is that the algorithm differs from one which decomposes the process of similarity calculation into a two-step process, where boundaries are first aligned ignoring boundary type, and then substitution costs are calculated. This is because substitution cost can affect whether a boundary in t_1 which corresponds to $\emptyset$ in t_2 is simply deleted, or transposed to match with a nearby boundary by the other annotator.

Our algorithm was implemented in R (R Core Team 2022). It takes input data formatted as an R `data.frame`, and outputs S_f and S_f^B, along with a record of each operation that took place. We additionally wrote a function to convert files in the .rez format imported from Rezonator (DuBois et al. 2020) into the required format for the function. The algorithm is available as an R package (https://github.com/rezonators/segsimflex).

3.3 Inter-annotator agreement

The similarity score measures how similar two annotations are, but how similar counts as 'good'? Converting the similarity to inter-annotator agreement (IAA) (Passonneau & Litman 1993, Hearst 1997) allows us to directly measure agreement among annotations. We use Cohen's κ, which compares the actual similarity between the annotations against chance-level similarity.

Following the definition of Cohen's κ, we calculate chance-level similarity based on the assumption that each boundary is a categorical random variable where each category is a boundary type or no boundary. The annotations are independent and identical within annotators and independent but non-identical across annotators. For example, with two boundary types p and q, the categories are $\{p, q, \emptyset\}$, and each annotator has their own $P(p)$, $P(q)$ and $P(\emptyset)$ values. Category probabilities are estimated with the maximum likelihood estimator, i.e. the proportion of that category within the annotation.

Based on these estimated null distributions, we then estimate chance-level similarity S_f^{chance} using the expected value of the similarity score. We use a simulation approach since it is difficult to find a closed form for it. At each simulation step, we draw a boundary type for each annotator at each boundary, then calculate the similarity score. The average similarity over k simulation steps is the estimated expected value of the similarity score. Cohen's κ is then calculated thus:

$$\kappa = \frac{S_f - S_f^{chance}}{1 - S_f^{chance}}$$

Hence, a negative score means below-chance performance, a positive score is above-chance, and perfect performance results in $\kappa = 1$.

Both S_f and S_f^B can be used for κ. If S_f is used, then the form of κ used here resembles the standard form of κ in classification tasks, except with gradient similarity between categories and an added possibility of transposition. Nevertheless, S_f may still be advisable at least in some situations (see Section 4.4 for discussion).

A common criticism of κ in classification contexts (Byrt, Bishop & Carlin 2010) is that large differences in raters' individual category distributions will deflate chance-level agreement and push κ up. In cases where this is expected to be a substantial problem, S_f^{chance} can instead be

calculated using an overall estimation of category probabilities that pools together both raters' annotations, turning the IAA into Scott's π.

Another common criticism is that a situation with unbalanced categories will lead to drastically higher expected proportion of agreement and thus lower κ values than one with balanced categories. This phenomenon is likely to occur with S_f-based κ, since non-boundaries are much more common than boundaries, but it is not necessarily problematic: A text with many non-zero boundaries is 'harder' to get right than a text with few non-zero boundaries, so if the aim is measuring rater performance (rather than the quality of the annotation itself), texts with more non-zero boundaries should have higher IAA than those with fewer non-zero boundaries but a comparable level of similarity. If the phenomenon *is* problematic, S_f^{chance} can instead be calculated based on the assumption that all boundary types (including no boundary) have equal probability, turning the IAA measure into Bennett's *S*. S_f^B-based κ ignores non-boundaries in normalising agreement, and thus is less likely to be subject to this phenomenon; if unevenness among boundary types is an issue, one may modify Bennett's *S* such that S_f^{chance} is calculated by getting a pooled estimate of the probability having no boundary from the two raters, then assuming the distribution of boundary types is uniform.

4 Case study: intonation unit segmentation

To illustrate the proposed measure, we apply our proposed measure to exploring inter-annoator agreement in a prosodic segmentation task.

4.1 Data and problem

We are manually segmenting the NCCU Taiwan Mandarin Corpus (Chui & Lai 2008) into intonational units (IUs), a unit of prosody corresponding to short bursts of speech (roughly corresponding to intonation phrases or breath groups in other prosodic frameworks). So far, we have annotated texts TM001, 004, 009, 016, 025, 036, 049. Before IU segmentation, we tokenised the texts to obtain potential boundary locations, following principles in Huang et al. (1997, 2017).

Two independent coders perform IU segmentation using four main boundary types, called *endnotes*, representing broad classes of prosodic contours near the end of the IU, each of which signals a type of transitional continuity (DuBois et al. 1993, DuBois 2020): Rising intonation indicating appeal, as in questions and uptalk (denoted by ?), continuing intonation indicating continuation of the prosodic sentence (a comma ,), falling intonation indicating finality (a period .), and a boundary marker for truncated IUs, i.e. IUs that ended before completion (a dash --). Some boundaries were uncategorised, usually because the IU consisted solely of elements with no discernable prosody, e.g. laughter or tsk-tsk; these are denoted as semicolon (;). Earlier on in the process, texts were segmented by manually editing text files; later, we performed segmentation using the Rezonator program (DuBois et al. 2020). Figure 3 shows the same tokens from one of the texts, TM001, as segmented differently by the two annotators who worked on this text. We calculate similarity scores and IAA on these texts to evaluate the quality of our annotation training and workflow and identify avenues for improvement.

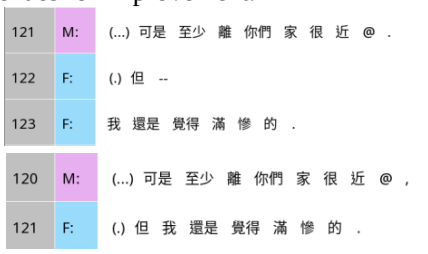


Figure 3: Example annotations in TM001. (a) and (b) are from two different annotators. The first boundary was deemed final by the first annotator, and continuing by the second. The word 但 *dàn* 'but' was put in a separate IU by the first annotator, but not the second.

4.2 Parameter values

For each pair of annotators, we calculated four values: S_f and S_f^B with an identity distance matrix, and the same values with the following custom similarity matrix:

$$M'_T = \begin{array}{c} \\ ? \\ , \\ . \\ -- \\ ; \\ \phi \end{array} \begin{array}{c} \begin{array}{cccccc} ? & , & . & -- & ; & \phi \end{array} \\ \begin{pmatrix} 1 & .5 & .25 & .25 & 1 & 0 \\ .5 & 1 & .5 & .5 & 1 & .25 \\ .25 & .5 & 1 & .25 & 1 & 0 \\ .25 & .5 & .25 & 1 & 1 & .25 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & .25 & 0 & .25 & 1 & 1 \end{pmatrix} \end{array}$$

Rising and falling intonation have the most dissimilar pitch contour of the four, hence a similarity of .25. Truncated intonation differs from

all others in not following a complete prosodic gestalt, and resembles continuing in having no rise/fall; hence the similarity with continuing is.5, and the similarity with the rest is .25. Rising and falling endnotes are substantially different intonationally from an IU-medial word, so their similarity with no boundary is 0; continuing and truncated IUs have less clear pitch cues and hence are harder to detect consistently, and receive .25 similarity. For simplicity, unclassified boundaries are ignored by treating them as identical to all other boundaries. Transposition costs are set at half the insertion/deletion cost for each endnote.

One may ask why we use these hand-crafted 'theoretical' values, instead of deriving values from empirical confusion matrices. This is because we want these values to reflect only difficulty in *prosodic* perception. However, actual boundary perception can be affected by grammatical structures derived from lexical content (Kuang et al. 2022). For example, in Hegemonic American English, statements often end with rises, and questions with falls (e.g. Bolinger 1999), and this is attested in our Mandarin data too. Though we tell annotators to consider only prosody, not content, they may still be affected by syntax and lexis, e.g. putting a question mark (?) after a syntactic/pragmatic question even though it has falling intonation. Such errors, even if common, need to be counted more heavily than errors caused by acoustic similarity. A possible alternative is to use confusion matrices from expert annotations assumed to *not* contain the syntax-based errors, which we do not pursue in this study because we do not yet have such datasets.

4.3 Results

Similarity scores are shown in Figure 4. As expected, I-based scores are lower than M'_T-based ones, and $S_f > S_f^B$ regardless of the similarity matrix, with S_f values nearing 1. The variation between texts is small within each measure, especially for S_f; there is greater variation in S_f^B.

The κ values are shown in Figure 5, where it is clear that S_f^B-based κ's remain substantially lower than S_f-based ones and I-based than M'_T-based ones. Overall, IAA scores are substantially lower than raw similarity scores, which is expected since they take into account the fact that chance-level similarity can be quite high. There is also less divergence between different measures for IAAs than raw similarities, suggesting that there is less difference as to how much each measure diverges from the chance-level value of that measure.

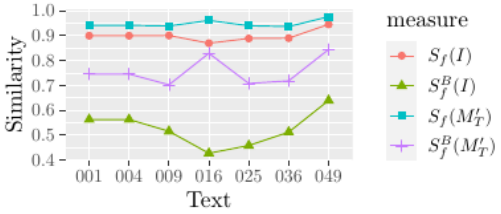


Figure 4: Various similarity metrics applied to texts

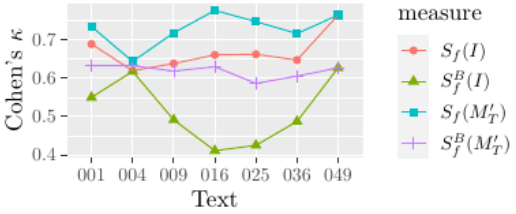


Figure 5: κ values for various similarity metrics.

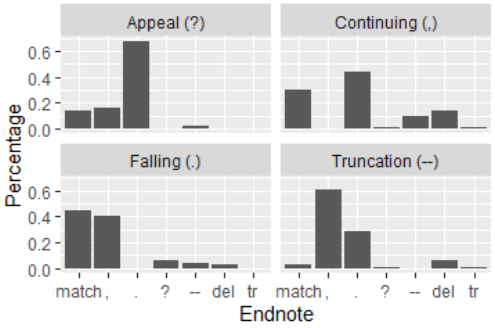


Figure 6: Distribution of operations performed on endnotes. 'Match' means full match, 'del' means deletion, 'tr' means transposition; the rest are substitutions between boundary types. Transposition plus substitution operations are not attested, and thus not shown.

Figure 6 shows the distribution of operations performed on each type of endnote in the annotations. The rate of full matches (i.e. both position and boundary type match) is quite low; falls are matched less than 50% of the time, the rest even less. Yet deletions and especially transpositions are rare, indicating high consistency for boundary *positions*: continuations have the most deletions, and even there the rate is less than 20%. Most of the errors are inconsistencies between boundary types. Truncations often correspond to continuations and sometimes to falls by other annotators. Falls and continuations are often confused for each other, while appeals correspond to falls around 70% of the time.

4.4 Discussion

The distributions of operations explain many of the patterns seen in the similarity score measures. Because most of the operations are substitutions between boundary types, once M'_T is used and correspondences between easily confusable boundary types are thereby downweighed, the similarity score rises drastically compared to I-based similarities. The dramatic disagreement with respect to boundary types may be attributable to a) lexical tone, which complicates perception as listeners must calibrate their perception of final pitch trajectories to the individual lexical tones; and b) the fact that words near IU boundaries, especially final particles, are often spoken very rapidly. Additionally, many appeal endnotes (?) were marked as falling (.) by the other annotator; manual inspection reveals some situations where the pitch contour is clear, but the one of the annotators decided between . vs ? based on syntax or pragmatics instead. Future annotator training will emphasise the importance of ignoring non-prosodic factors and calibrating intonational judgements according to lexical tones.

Notably, even when we consider Cohen's κ, a marked divergence between S_f and S_f^B remains. This is likely partially due to inherent weaknesses with using S_f^B for κ. In calculating chance-level similarity, the simulated annotations will have a comparable number of boundaries to the original annotations, because of how the distribution we simulate from is defined. But random placement of boundaries results in many mismatched boundaries, and hence a larger number of boundaries than actual annotations, which will have much more matches. This artificially inflates $S_f^{B,chance}$ compared to S_f^B, deflating S_f^B-based κ. Thus S_f may be the more suitable choice in κ calculation, and the moderate agreement indicated by S_f-based κ is a better indication of our annotation performance. This matches intuitively with the fact that boundary locations are mostly matched, while agreement on continuations and falls (the most common contours) are fair. The property of S_f^B discussed here may not have been noticed by Fournier (2013), who argued for *B* over *S*, because he focused on cases with full misses (insertion/deletion) and near-misses (captured by transpositions). He did not explore datasets like ours where *substitutions* between boundaries with largely matched positions are the primary operation.

Although we believe the *B*-based denominator is not optimal in this case, we do not claim that *N* is preferable in every scenario. For example, when one's main goal is to compare across texts to evaluate the difficulty of computationally detecting boundaries in each one, normalising with *N* unduly favours texts with sparser boundaries (longer segments). In ongoing work, we applied the measure to a case of evaluating a machine segmenter against different texts to determine the difficulty of segmenting different text types, and preliminary results show that S_f can give misleading results where S_f^B does not. We believe it is best to choose the denominator according to the specific dataset and problem.

5 Conclusion

In this paper, we introduced flexible segmentation similarity S_f, a new edit distance-based measure of segmentation similarity involving multiple mutually exclusive boundaries with fully flexible transposition, substitution, and addition/deletion costs. We justified its properties, presented an algorithm for computation, and extended it to inter-annotator agreement. We applied it to a case of intonation unit segmentation, where we evaluated consistency between manual segmentations and found ways to improve annotator training. We argued that, contrary to Fournier (2013), the number of boundaries is not always the best choice of denominator in calculating segmentation similarity for inter-annotator agreement when there is high agreement on boundary location but low agreement on boundary type. We hope our measure will find other use cases, especially where gradient differences between boundary types are needed.

Acknowledgements

Thanks to Lu Liu, Jack Sun, Sabrina Sun, Danni Wang, Sirui Wang, Haoran Yan and Sunny Zhong for their annotation work, John W Du Bois for valuable guidance throughout the project, Haoran Yan, Olivia Jonokuchi, Laurel Brehm, Simon Todd and UCSB's CEILing group for comments on an earlier draft, Sherry Chien for her involvement early in the project, and Tianrui Gu and Giselle Ramirez for their current work extending the package.

References

Barth-Weingarten, Dagmar. 2016. *Intonation units revisited: cesuras in talk-in-interaction* (Studies in Language and Social Interaction 29). Amsterdam ; Philadelphia: John Benjamins Publishing Company.

Beeferman, Doug, Adam Berger & John D. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning* 34(1–3). 177–210.

Byrt, Ted, Janet Bishop & John B. Carlin. 1993. Bias, prevalence and kappa. *Journal of Clinical Epidemiology* 46(5). 423–429. https://doi.org/10.1016/0895-4356(93)90018-V.

Chui, Kawai & Huei-ling Lai. 2008. The NCCU corpus of spoken Chinese: Mandarin, Hakka, and Southern Min. *Taiwan Journal of Linguistics* 6(2).

DuBois, John W., Schuetze-Coburn, Susanna Cumming & Danae Paolino. 1993. Outline of discourse transcription. (Ed.) Jane A. Edwards & Martin D. Lampert. *Talking data: Transcription and coding in discourse research*. Lawrence Erlbaum Associate, Inc. Publishers.

DuBois, John W. 2020. Discourse Functional Transcription: Conventions. Unpublished manuscript.

DuBois, John W., Terry DuBois, Georgio Klironomos & Brady Moore. 2020. From answer to question: Coherence analysis with Rezonator. In *Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue*.

Fournier, Chris. 2013. Evaluating text segmentation using boundary edit distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1702–1712.

Fournier, Chris & Diana Inkpen. 2012. Segmentation Similarity and Agreement. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics*.

Franz, Martin, J. Scott McCarley & Jian-Ming Xu. 2007. User-oriented text segmentation evaluation measure. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 701–702.

Huang, Chu-Ren, Keh-Jiann Chen, Li-Li Chang & Feng-Yi Chen. 1997. Segmentation standard for Chinese natural language processing. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 2, Number 2, August 1997*, 47–62.

Huang, Chu-Ren, Shu-Kai Hsieh & Keh-Jiann Chen. 2017. *Mandarin Chinese words and parts of speech: A corpus-based study*. Routledge.

Kuang, Jianjing, May Pik Yu Chan & Nari Rhee. 2022. The effects of syntactic and acoustic cues on the perception of prosodic boundaries. *Proc. Speech Prosody 2022* 699–703.

Lin, You-Jing. 2009. *Units in Zhuokeji rGyalrong discourse: Prosody and grammar*. University of California, Santa Barbara.

Lu, Wei & Hwee Tou Ng. 2010. Better punctuation prediction with dynamic conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, 177–186.

Passonneau, Rebecca J. & Diane J. Litman. 1993. Intention-based segmentation: Human reliability and correlation with linguistic cues. *Proceedings of ACL-93*.

Peshkov, Klim & Laurent Prévot. 2014. Segmentation evaluation metrics, a comparison grounded on prosodic and discourse units. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Peshkov, Klim, Laurent Prévot & Roxane Bertrand. 2013. Evaluation of automatic prosodic segmentations. In *Interface Conference 2013 (IDP-2013)*, 95.

Pevzner, Lev & Marti A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics* 28(1). 19–36.

Qian, Peng, Xipeng Qiu & Xuan-Jing Huang. 2016. A new psychometric-inspired evaluation metric for Chinese word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), 2185–2194.

R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Troiani, Giorgia, DuBois, John W. & Gries, Stefan Th. (2023). Testing the perception of Intonation Unit boundaries in naturally occurring conversation. XV International Conference on General Linguistics, University of Madrid Complutense.

Assessing the featural organisation of paradigms with distributional methods

Olivier Bonami
Université Paris Cité,
Laboratoire de linguistique formelle,
CNRS
olivier.bonami@u-paris.fr

Lukáš Kyjánek
Charles University, Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
kyjanek@ufal.mff.cuni.cz

Marine Wauquier
Université Sorbonne Nouvelle,
Laboratoire Lattice,
CNRS
marine.wauquier@sorbonne-nouvelle.fr

Abstract

In this paper, we apply distributional methods to Czech data to compare the predictions of two views of inflectional paradigms, as systems of orthogonal morphosyntactic feature oppositions, or as systems of multilateral contrasts between pairs of morphologically related words, not necessarily reducible to orthogonal features.

We define two predictive tasks that probe what it means for two pairs of paradigm cells to contrast in the same features: in the first, we train a classifier to discriminate between two paradigm cells; in the second, we train a family of models to predict the vector of the word in one cell from that of the word in another cell. By varying the choice of training and test data, we show that (i) a model trained on data that contrast in a manner orthogonal to its test data performs on average at chance level, while (ii) a model trained on data that contrast in a manner parallel to its test data performs on average better than chance but still worse than a model trained on the same pair of cell used for testing. This is incompatible with the predictions of a reductive view of paradigms as systems of feature contrasts.

1 Introduction

The notion of an inflectional paradigm is an invaluable tool for linguistic description and has played an increasing role in linguistic theory in the last few decades. Explicit reference to paradigm structure has been claimed to be necessary to account for phenomena as diverse as patterns of syncretism (Zwicky, 1985; Stump, 1993; Baerman et al., 2005), competition between synthetic and periphrastic expression of morphosyntactic categories (Ackerman and Stump, 2004; Kiparsky, 2005; Bonami, 2015), and universal constraints on the shape of inflection systems (Carstairs-McCarthy, 1994; Ackerman and Malouf, 2013). While many of these claims have been met with scepticism by some (see e.g. papers collected in Bachrach and Nevins 2008), there is general agreement that some form of paradigmatic organisation plays a role in morphology, if only through the existence of collections of pairs of expressions that differ by contrasting in the same morphosyntactic features. Hence although morphologists may differ in how they think of paradigms, they will agree that there is something in common between the way *man* relates to *men* and *dog* relates to *dogs*. That something in common is what we will call a paradigmatic relation.

That being said, there is variation in the literature regarding the way paradigms are defined, and differences between these formulations are seldom discussed. A common position, ultimately grounded in Jakobson (1958) and cogently articulated by Wunderlich and Fabri (1995, p. 266), holds that "A paradigm is an n-dimensional space whose dimensions are the attributes (or features) used for the classification of word forms". In other words, paradigms can be reduced to a system of orthogonal contrasts in morphosyntactic feature values.[1] This claim is appealing when we look at some very-well behaved inflection systems. Consider the paradigm of an Italian adjective in Table 1. Every cell in that paradigm can be defined as the combination of a number and a gender value. If this holds in general, it suggests that paradigm structure is entirely

[1] Note that we follow Matthews (1991) in calling 'morphosyntactic' whatever features are relevant to the organisation of inflectional paradigms. Some of these will be semantically relevant, others not. Our usage departs from that of Corbett (2012), who would call some of the features we discuss here 'morphosemantic'.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 310-320.
Amherst, Massachusetts, June 15-17, 2023

	MAS	FEM
SG	buono	buona
PL	buoni	buone

Table 1: Paradigm of Italian BUONO 'good'.

			IND PRS	IND PST	IMP
FINITE	SG	1	eat	ate	—
		2	eat	ate	eat
		3	eats	ate	—
	PL	1	eat	ate	—
		2	eat	ate	eat
		3	eat	ate	—
NFIN	PART		eating	eaten	
	INF		eat		

Table 2: Paradigm of English EAT as a system of orthogonal oppositions. Periphrastic forms ignored.

derivative of a system of feature oppositions.

This view of paradigms becomes less appealing as soon as we move away from well-behaved declension systems. In conjugation systems, it often is the case that orthogonal feature oppositions are unhelpful. English conjugation provides an extreme example of that situation. Table 2 is our best attempt at presenting the paradigm of an English verb as a system of orthogonal oppositions. Multiple problems arise: some feature oppositions are neutralised (no tense distinction in the imperative or infinitive), and some paradigm cells are non-existent (no 1st or 3rd person imperatives). Most importantly, there is a disconnect between the shape of the paradigm as motivated by feature oppositions and the inventory of forms filling that paradigm: with the exception of BE, no lexeme uses more than 5 distinct forms to fill 17 cells, and arbitrary collections of cells exhibit systematic syncretism — e.g. all non-3rd present form, imperative forms, and the bare infinitive.

The observation of such discrepancies naturally leads one to revise their expectations as to the paradigmatic organisation. Spencer (2013), Boyé and Schalchli (2016), and Stump (2016) make slightly different proposals for distinguishing different notions of paradigms. Bonami and Strnadová (2019), building among others on Štekauer (2015) and Blevins (2016, chap. 5), take another route illustrated for English verbs in Figure 1. Under this view, contrasts in content between sets of pairs of words, materialised in the figure by vertical alignments across morphological families, are the

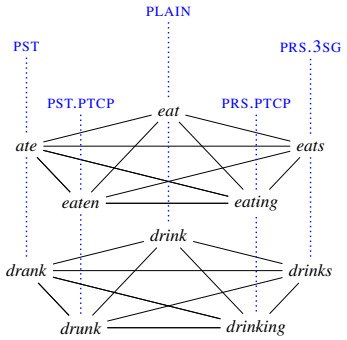


Figure 1: English verbal paradigms seen as a system of basic contrasts in content.

primitive notion from which paradigms are defined. Analysis of such paradigms in terms of orthogonal features is a further step that may be more or less useful and insightful depending on the system under examination. Crucially, paradigms (horizontal planes in Figure 1) and paradigm cells (vertically aligned collections of words) exist independently of such a featural analysis.

In this paper, we explore empirically the predictions of the two basic conceptualisations of paradigm structure outlined above. Focusing on cases where a feature-based definition of paradigms seems warranted as in Table 1, we ask to what extent the featural composition of the paradigm can be trusted. For example, is the contrast between masculine singular and plural really the same as the contrast between feminine singular and feminine plural? To answer that question, we explore contrasts between pairs of words (nouns or adjectives) in Czech using distributional vectors familiar from distributional semantics. Note that distributional vectors typically capture both syntactic and semantic contrasts between words. While this is sometimes an embarrassment when disentangling the two is important, it is fine for our purposes, as paradigmatic contrasts may be semantically potent or not.

Section 2 provides a precise definition of what it means for two pairs of cells to encode contrasts that are parallel, orthogonal or neither. We then use this definition to lay out predictions on the expected structure of the distributional vector space under the assumption that paradigms are defined by features. In Section 3 we present two experiments testing these predictions: in the first experiment, we train classifiers to discriminate between vectors of words from two paradigm cells, while in the

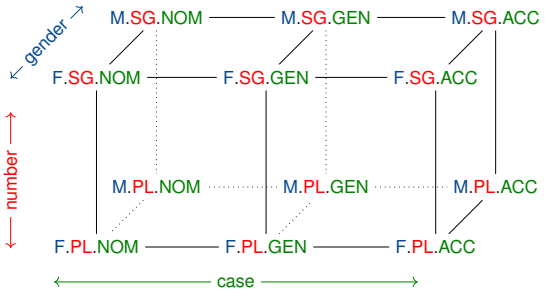


Figure 2: Illustrative organisation of a paradigm as a system of orthogonal featural contrasts. In this example, we have three features, namely case, number and gender, represented as three geometric dimensions. Paradigm cells are represented as points in 3D space combining a particular value for each feature.

second experiment, we train a model to predict the vector of a word in one paradigm cell from that of the word in another paradigm cell. In both cases, we compare the quality of prediction of models trained on data from the same pair of cells, from a parallel pair of cells, or from an orthogonal pair of cells. Section 4 discusses the implications of our findings for morphological theory, and Section 5 outlines avenues for future work.

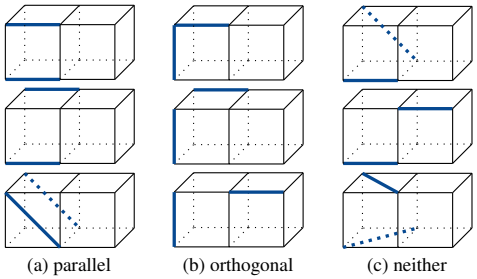
This paper presents a terminological difficulty, as the term 'feature' has different meanings in the context of descriptive and theoretical morphology and in the context of computational linguistics and machine learning. To alleviate that difficulty, we refrained from using the term at all when discussing machine learning, talking of *predictors* or *variables* instead; and we prefixed *feature* with *morphosyntactic* wherever there was potential for ambiguity.

2 Predictions

In this section, we define ways of comparing how inflected forms of the same lexeme differ in meaning and use this to derive predictions of the claim that paradigms reduce to featural contrasts.

For the sake of exploring the featural organisation of paradigms, we assume that each cell in a paradigm can meaningfully be mapped to a morphosyntactic description which we formalise as a functional relation between a set of features $\mathcal{F}$ and a set of values $\mathcal{V}$, where no two features can map to the same value.[2] Given two paradigm

[2]We follow Stump and Finkel (2013) in assuming that the list of paradigm cells can be a proper subset of the set of all such functional relations, leaving room for the description of systems such as that exemplified with English conjugation above. The requirement that no two feature map to the same values is purely motivated by mathematical elegance, and could easily be dropped.



(a) parallel (b) orthogonal (c) neither

Figure 3: Types of relations between pairs of cells.

cells a and b, we note $S(a,b) \stackrel{def}{=} \{v \mid f : v \in a \wedge \neg f : v \in b\}$ the set of feature values specific to a when compared to b. We then say that two pairs of contrasting cells (a,b) and (a',b') are **parallel** if $S(a,b) = S(a',b')$ and $S(b,a) = S(b',a')$. We likewise note $C(a,b) \stackrel{def}{=} \{f|\exists v \exists w[f : v \in a \wedge f : w \in b \wedge v \neq w]\}$ the set of features along which a and b contrast, and then call two pairs of cells **orthogonal** if they do not share any contrast, i.e. $C(a,b) \cap C(a',b') = \emptyset$.

For purposes of illustration, we will use the example laid out in Figure 2 of a system with two binary features (number and gender) and one ternary feature (case), and represent visually each feature as a geometric dimension. The definitions of parallelism and orthogonality are illustrated in Figure 3. Note that we can find parallel pairs of contrasts where the contrasting cells have no feature in common (bottom left). Note also that our notion of parallelism does not extend to situations where the two contrasts involve the same features but different values (middle right): in that situation, contrasts are neither parallel nor orthogonal.

Given these definitions, we can now derive our predictions. Let us assume that we have satisfactory representations of the content of inflected words in a language (combining semantic and syntactic information). Let us also assume that paradigmatic relations are fully reducible to some correct description in terms of feature contrasts. Then, given two pairs of words (v,w) and (v',w') filling cells (a,b) and (a',b') of some paradigm:

- If (a,b) and (a',b') are parallel, then the content of v and w should differ in exactly the same way as the content of v' and w' differ. Hence if we define a predictive task which relies on capturing the relationship between

cells a and b, it should be immaterial whether we train our system on data from cells a and b (what we call *intrinsic prediction*) or cells a' and b' (what we call *extrinsic prediction*).

- If (a, b) and (a', b') are orthogonal, then the contrast between the content of v and w is unrelated to the contrast between the content content of v' and w'. Hence if we define a predictive task which relies on capturing the relationship between cells a and b and train our system on data from cells a' and b', we should witness dramatically poor performance, at the chance level.

In Section 3, we test these predictions on data from Czech nouns and adjectives. Czech nouns inflect for 2 numbers (singular, plural) and 7 cases (nominative, genitive, dative, accusative, vocative, locative, instrumental), leading to a 2-dimensional system with 14 cells, while adjectives also inflect for 4 genders (masculine animate, masculine inanimate, feminine, neuter) and 3 grades (positive, comparative, superlative), leading to a 4-dimensional system with 168 cells. In the interest of tractability, we restrict attention to the positive grade of adjectives and the three main structural cases (nominative, genitive, accusative). This leads for nouns to 6 cells in 2 dimensions, and for adjectives to 24 cells in 3 dimensions — see Tables 3 and 4 for examples. We also leave out from consideration orthogonal contrasts forming a corner, as in the top example of column (b) in Figure 3, as sharing of a cell between the two pairs is likely to affect performance.

3 Experiments

3.1 Data

We use distributional representations of Czech word vectors from the vector spaces provided by Kyjánek and Bonami (2022). These models were trained by applying word2vec (Mikolov et al., 2013) to the SYN v9 corpus (Křen et al., 2021), a large corpus of contemporary edited text compiled, lemmatised and tagged by the Czech National Corpus team (4,719M tokens; 7.3M lemmas; 362M sentences). Vectors were trained on the concatenation of tokens and POS tags, and hence in effect represent a form filling a particular paradigm cell. For instance FEM.NOM.SG and NEU.NOM.PL *malá* from Table 3 get separate representations. This is crucial for our purposes:

POSITIVE GRADE

		MA	MI	FEM	NEU
SG	NOM	**malý**	**malý**	**malá**	**malé**
	GEN	**malého**	**malého**	**malé**	**malého**
	DAT	malému	malému	malé	malému
	ACC	**malého**	**malý**	**malou**	**malé**
	VOC	malý	malý	malá	malé
	LOC	malém	malém	malé	malém
	INS	malým	malým	malou	malým
PL	NOM	**malí**	**malé**	**malé**	**malá**
	GEN	**malých**	**malých**	**malých**	**malých**
	DAT	malým	malým	malým	malým
	ACC	**malé**	**malé**	**malé**	**malá**
	VOC	malí	malé	malé	malá
	LOC	malých	malých	malých	malých
	INS	malými	malými	malými	malými

COMPARATIVE GRADE

		MA	MI	FEM	NEU
SG	NOM	menší	menší	menší	menší
	GEN	menšího	menšího	menší	menšího
	DAT	menšímu	menšímu	menší	menšímu
	ACC	menšího	menší	menší	menší
	VOC	menší	menší	menší	menší
	LOC	menším	menším	menší	menším
	INS	menším	menším	menší	menším
PL	NOM	menší	menší	menší	menší
	GEN	menších	menších	menších	menších
	DAT	menším	menším	menším	menším
	ACC	menší	menší	menší	menší
	VOC	menší	menší	menší	menší
	LOC	menších	menších	menších	menších
	INS	menšími	menšími	menšími	menšími

SUPERLATIVE GRADE

		MA	MI	FEM	NEU
SG	NOM	nejmenší	nejmenší	nejmenší	nejmenší
	GEN	nejmenšího	nejmenšího	nejmenší	nejmenšího
	DAT	nejmenšímu	nejmenšímu	nejmenší	nejmenšímu
	ACC	nejmenšího	nejmenší	nejmenší	nejmenší
	VOC	nejmenší	nejmenší	nejmenší	nejmenší
	LOC	nejmenším	nejmenším	nejmenší	nejmenším
	INS	nejmenším	nejmenším	nejmenší	nejmenším
PL	NOM	nejmenší	nejmenší	nejmenší	nejmenší
	GEN	nejmenších	nejmenších	nejmenších	nejmenších
	DAT	nejmenším	nejmenším	nejmenším	nejmenším
	ACC	nejmenší	nejmenší	nejmenší	nejmenší
	VOC	nejmenší	nejmenší	nejmenší	nejmenší
	LOC	nejmenších	nejmenších	nejmenších	nejmenších
	INS	nejmenšími	nejmenšími	nejmenšími	nejmenšími

Table 3: Paradigm of Czech MALÝ 'small'. Cells used in the experiments are highlighted in boldface.

	SG	PL
NOM	**holka**	**holky**
GEN	**holky**	**holek**
DAT	holce	holkám
ACC	**holku**	**holky**
VOC	holko	holky
LOC	holce	holkách
INS	holkou	holkami

	SG	PL
NOM	**cíl**	**cíle**
GEN	**cíle**	**cílů**
DAT	cíli	cílům
ACC	**cíl**	**cíle**
VOC	cíli	cíle
LOC	cíli	cílech
INS	cílem	cíli

Table 4: Paradigms of two Czech nouns: feminine HOLKA 'girl' and masculine inanimate CÍL 'goal'. Cells used in the experiments are highlighted in boldface.

since syncretism is rampant in Czech inflection, distributional representations of raw strings would be useless to make comparisons across paradigm cells. We used the tagging distributed with the corpus, which was obtained automatically using the MorphoDiTa tool (with a reported accuracy over 95%, Straková et al., 2014). In our experiments, we use a 100-dimensional vector space trained as a continuous bag of words (CBOW) model.[3] We also used the inflectional morphological dictionary MorfFlexCZ 2.0 (Hajič et al., 2020), which contains 125.3M triplets of word form and its respective lemma and tag, to sample vectors of tokens with relevant morphosyntactic categories. Note that MorfFlexCZ and the SYN corpus share the same tagset.

For the first experiment, we sampled 500 random word vectors for each paradigm cell under investigation, allowing us to have combined datasets for classification of size 1000. We included only word vectors for words that occurred at least 50 times in the SYN v9 corpus. This led to 24 datasets for adjectives corresponding to the 24 paradigm cells highlighted in Table 3. For nouns we created separate datasets for each of the genders, leading again to 24 ($= 4$ genders $\times$ 6 paradigm cells) datasets.

For the second experiment, we needed datasets consisting of ordered pairs of vectors for forms of the same lexeme for two particular cells in the paradigm. We used MorfFlexCZ to identify relevant pairs and randomly sampled datasets of 1,000 pairs; again, we included only vectors with a frequency of 50 or more. For adjectives, with 24 paradigm cells under examination, we ended up with $24 \times 23 = 552$ datasets. For nouns, we again created separate datasets for each gender. With 6 paradigm cells under examination, this led to $4 \times 6 \times 5 = 120$ datasets.

3.2 Experiment 1

In our first experiment, we want to assess how hard it is to discriminate two paradigm cells when trained on data from the same or other cells. To this end, we train classifiers to discriminate between two paradigm cells and apply it to data from the same pairs of cells, parallel pairs of cells, and orthogonal pairs of cells.

More specifically, we design two-step experiments. First, we conduct *intrinsic classification*, meaning that we train a classifier to discriminate a given contrast realised by a pair of paradigm cells, and we apply it to words inducing the same contrast. An example of this would be training to discriminate FEM.SG.ACC and FEM.PL.ACC forms of adjectives, and testing the classifier on the forms of other lexemes in the same two cells. Second, we investigate the interoperability of the morphosyntactic feature by means of an *extrinsic classification* task. An example of this would be training to discriminate FEM.SG.ACC and FEM.PL.ACC forms of adjectives, and testing the performance of the classifier on its ability to discriminate words in two other cells, e.g. FEM.SG.GEN and FEM.PL.GEN. We hypothesise a classifier trained to discriminate the contrast between two cells should also be able to discriminate between two other cells provided the two pairs of cells are parallel.

Concretely, for each relevant predictor pair of cells (a, b), we train a classifier to discriminate vectors of words in cell a from vectors of words in cell b. We used gradient boosting (Friedman, 2001a; Mason et al., 2000) applied to decision trees as our classification method. Predictors are the 100 dimensions of the vectors, and boosting trees parameters are set to 500 estimators, a learning rate of 0.01, a max depth of 2, a random state of 0, and the deviance loss function. In total, we trained 60 classifiers for nouns, to be used in 60 and 86 intrinsic and extrinsic classification tasks respectively; and 276 classifiers for adjectives, used in 276 intrinsic and 7824 extrinsic classifications tasks. The much higher number of tasks for adjectives is due to their larger paradigm size due to gender agreement, cf. Tables 3 and 4.

For intrinsic classification tasks, we performed 10-fold cross-validation, and report aggregated accuracy across the 10 folds. For extrinsic classification, there was no avoidable risk of over-fitting, as the training and test datasets are inherently disjoint.[4] Note that, since our samples are balanced, chance performance is at 0.5. We use this as our baseline for evaluation. Figure 4 summarises our results.

Classifiers for both nouns and adjectives achieve very high performance at intrinsic classification,

[3] We also experimented with models trained by the skip-gram method or having 400-dimensional vectors, but this led to no qualitative difference in the results.

[4] As a reviewer notes, the test data is included in the training corpus for the vector space, and hence can in principle have some influence on the results. There is no way of avoiding that potential problem with the methods used here, as we do need vectors from the same space for test items for evaluation purposes.

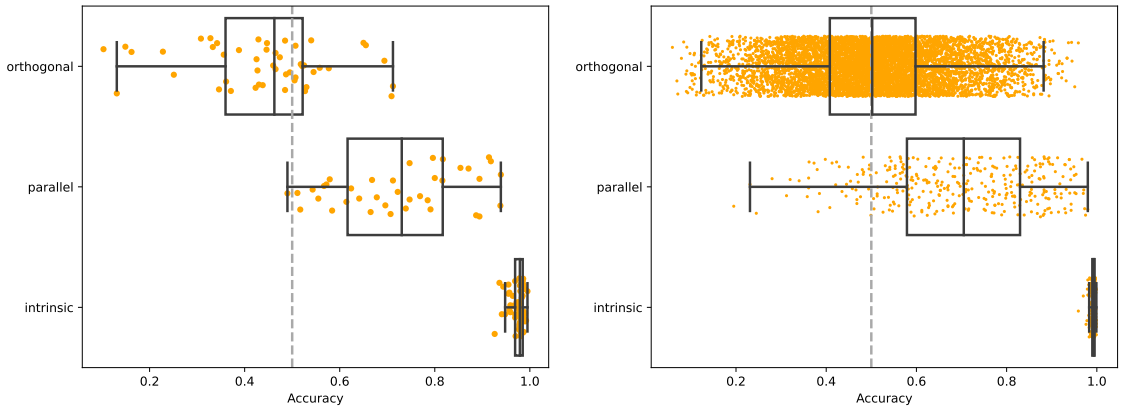


Figure 4: Distribution of accuracy of classifications (Experiment 1) for nouns (left) and adjectives (right). The dashed grey line represents baseline performance at 0.5.

with a median accuracy of 0.98 and 0.99 respectively and a standard deviation of 0.02 and 0.005 respectively. Performances are significantly lower for extrinsic classification, although the use of classifiers for parallel contrasts still leads to above-chance level performance for a vast majority of models, with a median accuracy of 0.72 for nouns and 0.71 for adjectives. On the other hand, extrinsic classification for orthogonal contrasts barely achieves chance-level performances. Median accuracy is at 0.46 for nouns and 0.51 for adjectives. There is a lot of variation around this median, which is not surprising given the high number of models we trained, but the distribution of accuracy across orthogonal classifiers is clearly symmetric and centred on 0.5, suggesting that any structure that individual classifiers pick out is due to lucky sampling.

3.3 Experiment 2

In our second experiment, we predict the vector of a word in the target paradigm cell ($\vec{v}_{\text{predicted}}$) from that of the word in another paradigm cell ($\vec{v}_{\text{predictor}}$), and evaluate the quality of our prediction by comparing it to the actual vector $\vec{v}_{\text{actual}}$. This is represented graphically in Figure 5, where $\mathcal{M}$ denotes the model deriving the prediction.

Multiple ways of constructing the model $\mathcal{M}$ are found in the literature. A simple approach relies on adding to the predictor vector the offset vector relating two words standing in the same relation (Mikolov et al., 2013) or averaging over such offset vectors (Drozd et al., 2016; Mickus et al., 2019). Marelli and Baroni (2015) propose instead to use a linear transformation to predict the target vector

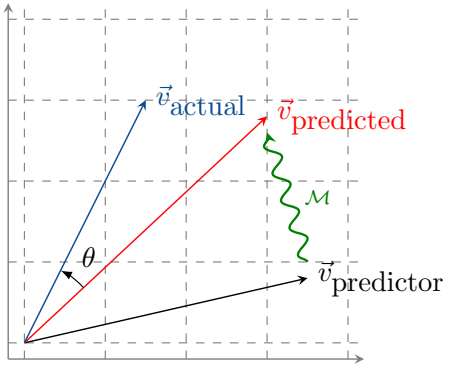


Figure 5: Evaluation of vector prediction. Performance of model $\mathcal{M}$ is assessed by the cosine of the angle θ between the actual vector for the target word and the vector predicted by $\mathcal{M}$ for the that based on the predictor vector of a related word.

— that is, they predict the value of each dimension of the target vector using a linear combination of the values of all dimensions in the predictor vector. They argue that this should allow capturing at least some aspects of affix polysemy. Bonami and Naranjo (2023) use a variant of this approach using principal component analysis to reduce the number of independent variables in the linear models.

In this paper we follow closely the methodology of Marelli and Baroni (2015), using Gradient Boosting Tree regression models (Friedman, 2001b) instead of linear models.[5] For each morphological contrast, we train 100 models per pairwise combination of paradigm cells as there are 100 vector dimensions in the input vector space models. In to-

[5]We also tested linear regression models, but the gradient boosting tree method achieved better evaluation results.

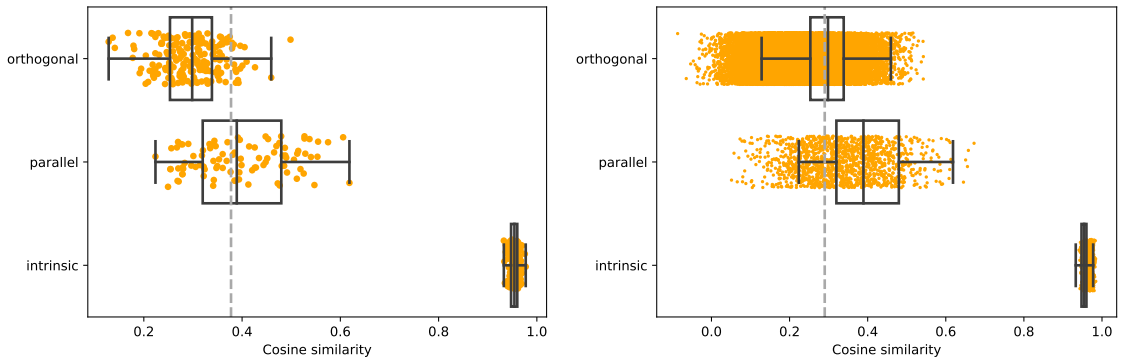


Figure 6: Distribution of quality (cosine similarity) of vector predictions (Experiment 2) for nouns (left) and adjectives (right). Grey lines indicate the average cosine similarity between members of the same lemma.

tal, we trained $100 \times (120+552) = 67,200$ models ($\times 10$ because of cross-validation) to predict all vector dimensions of words from the paradigm cells under analysis. We then evaluate the performance of our models in both intrinsic and extrinsic predictions, using the average cosine similarity between predicted and actual vector ($\cos(\vec{v}_{\text{predicted}}, \vec{v}_{\text{actual}})$) as our measure of quality. While the evaluation of the intrinsic predictions assesses discriminating power for predicting word vectors, i.e., the prediction of the same contrasts as the one on which the model was trained, the evaluation of the extrinsic predictions assesses the stability of predicting word vectors in different contexts, i.e., the prediction of contrasts different from the one used for training the model.

Results are presented in Figure 6. We get very high scores for intrinsic prediction, ranging between 0.92 and 0.98. Cross-validated models have barely lower performance (median difference 0.012, max. 0.02), indicating that there is no over-fitting to speak of. The extrinsic predictions achieved vastly lower cosine similarities than their intrinsic counterparts, with a gap of more than 25% between the best-performing extrinsic prediction and the worst-performing intrinsic prediction. As in Experiment 1, results for both orthogonal and parallel prediction are quite spread out, but there is a clear central tendency to have higher performance for parallel prediction than for orthogonal prediction.

We contextualise the results of trained models in two ways. Our first approach is to compute the average pairwise cosine similarity between vectors of words belonging to the same lemma, for the paradigm cells of interest, and for each part of speech. This gives us an indication of what would be the performance of a model that perfectly captured the fact that the target vector conveys the right lexical semantics, but does not capture anything about the contribution of morphosyntactic features. These are materialised by grey lines in Figure 6. It is most relevant to compare that number to the performance of intrinsic models: here we see very clearly that these models do capture much more than just the lexical semantics associated with belonging to the same lemma.

For orthogonal and parallel prediction, this comparison is hard to interpret, given the high variability of the quality of prediction across tasks of the same type. We suspect that this variability is due at least in part to the fact that some test sets are inherently easier or harder to predict due to the structure of the vector space. We hence develop a baseline that is directly sensitive to the test set, and we compare the results of our cross-validated models to those from the baseline. The simplest baseline would be to create a predicted vector from random numbers; however, sampling random numbers might lead to vectors that are out of the vector space model. Therefore, we instead pick random word vectors from the vector space model and use them as predicted word vectors. To mitigate knowledge that such randomly picked word vectors might encode, we pick randomly 20 word vectors for each pair of word vectors and calculate the average of cosine similarities between the actual vector $\vec{v}_{\text{actual}}$ and individual randomly picked word vectors ($\vec{v}_{\text{predicted}_1}, \ldots, \vec{v}_{\text{predicted}_{20}}$). The resulting cosine similarity for a given contrast is computed as the average of the averages achieved by individual pairs of word vectors.

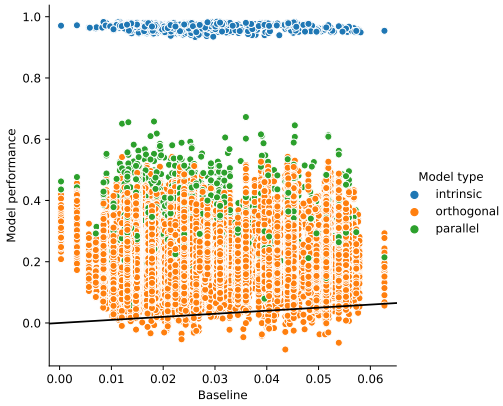


Figure 7: Comparison of our models to the baseline. The black line stands for equal values on the x and y axis.

Figure 7 shows pairwise comparisons between baseline and model performance. The clear conclusion is that both intrinsic and parallel prediction clearly outperform the baseline. A few orthogonal models perform at the baseline level, but most still clearly beat the baseline. To put these results in perspective, it is important to remember that, while orthogonal models are trained on irrelevant morphosyntactic contrasts, they still see pairs of forms of the same lexeme. To the extent that the vectors disentangle lexical semantics from morphosyntactic features, they should still be able to predict lexical semantics correctly — by not changing the values of the relevant dimensions. It is hence expected that performance should be above baseline on average; the fact that it is not always suggested that lexical semantics and morphosyntactic features are not clearly separated by the vectors.

4 Discussion

Our two experiments lead to similar results that we discuss in the following paragraphs.

First, intrinsic prediction works very well: classifiers learning to discriminate two paradigm cells on the basis of the corresponding word vectors reach very high accuracy, even under cross-validation; and a model learning to deduce the vector in one cell from the vector in another cell makes predictions that are very close to the actual vectors, and go well beyond capturing the fact that words belonging to the same lemma tend to be similar. Together, these indicate that the word vectors we use do capture the relevant syntactic and semantic differences between paradigm cells with a high degree of accuracy.

Second, orthogonal prediction leads to poor performance: training a model on a contrast orthogonal to that found in the test data is, unsurprisingly, a bad idea. This is most clearly established for the classification task of Experiment 1, where we see that most models have a performance close to the baseline, while a few models got lucky or unlucky, in a symmetric fashion. In the vector prediction task of Experiment 2, performance is still on average much better than the random baseline, due to the fact that orthogonal models, unlike the baseline, have the capacity to accurately predict some aspects of distributions that are due to being forms of the same lexeme.

The third and most important result is that found in the situation of parallel prediction, where a model is trained on one pair of cells implementing a feature contrast and tested on a different pair of cells implementing the same feature contrast. Here we find that, in both experiments, performance is measurably higher (on average) than with orthogonal models, but markedly lower than in intrinsic prediction. This last result is in direct contradiction to the predictions laid out in Section 2. If contrasts between paradigm cells were fully reducible to contrasts in feature values, then parallel pairs of cells should contrast in exactly the same way, and hence parallel prediction and intrinsic prediction should lead to comparable performance.

These results lead to a nuanced view of the role of morphosyntactic features in the analysis of inflectional paradigms. First, paradigm structure is not fully reducible to a system of orthogonal feature contrasts, *pace* Wunderlich and Fabri (1995) and many others. Paradigm cells have irreducible distributional properties that cannot be deduced from their featural analysis. Note that this is compatible with the view articulated by Bonami and Strnadová (2019), where each paradigm cell is characterised by the full set of its contrasts with all other cells. Second, morphosyntactic features do capture relevant similarities between pairs of cells: if they did not, parallel predictions should fare no better than orthogonal predictions.

Of course, one may dispute the extent to which these results are relevant to the featural analysis of paradigms. Our results are compatible with a situation where distributional vectors are influenced by morphosyntactic features, which are nicely organised in orthogonal dimensions, plus some other

factors, which are not. We see no empirical way of dismissing such an analysis. However, we submit that it does not affect our conclusion: whatever the relevant factors are, it remains that paradigm cells have properties that are not reducible to orthogonal features.

Let us finish by noting that the nuanced conclusion (features capture some but not all paradigm structure) is most congruent with what Blevins (2006) calls an abstractive model of morphology. Under this view, surface words and the surface relations they entertain are the basic primitive, and objects such as stems and affixes are abstractions that may (but need not) be defined out of words and their relations. Arguably, morphosyntactic features can also be seen as such useful abstractions, that do not *define* paradigmatic relations but highlight some of their properties.

5 Outlook

We end by discussing areas of future research based on the results presented in this paper.

First, this paper did not explore what it is exactly that makes contrasts across parallel pairs of paradigm cells different; for instance, we did not look into whether some feature contrasts are easier or harder to predict, or more or less parallel across pairs of cells. We leave such questions for future research. We also leave for the future detailed analysis of particular parallel contrasts: we could e.g. examine distributional similarities and differences for a set of nouns in the NOM.SG, ACC.SG, NOM.PL and ACC.PL, and see whether these explain the performance models on this particular set of contrast.

Second, we focused in this paper on cases where the assumption of orthogonality of features was maximally convincing. A different use of the same methodology would be to explore situations where the literature is disputed as to what the feature contrasts actually are and attempt to settle the dispute by assessing how fruitful a feature analysis is in terms of capturing distributional parallelism or orthogonality. Obvious targets include Jakobson (1958)'s three-dimensional analysis of the Russian case systems, as well as many later proposals inspired by it; or the vexed question of the independence of person and number (see e.g. Siewierska 2004).

Third and finally, we have not explored whether and how different morphosyntactic features differ in their degree of parallelism across contrasts. We have reasons to believe that they could. Much recent literature has highlighted the multidimensional and gradient nature of the distinction between inflection and derivation (Booij, 1996; Bauer, 2004; Corbett, 2010; Spencer, 2013); in particular, semantically potent inherent morphosyntactic features, such as the number of nouns, are more derivation-like that purely morphosyntactic and contextual features, such as grammatical case. Previous research has shown that inflectional and derivational morphological relations as a whole difference in the predictability of their distributional consequences (Bonami and Paperno, 2018), and found some distributional reflexes for the existence of a gradient (Copot et al., 2022). Degree of parallelism might be another relevant distributional property: we may expect, for instance, there to be less parallelism of the number feature across cases than of cases across the number feature.

Acknowledgements

We thank Sacha Beniamine, Mae Caroll, Maria Copot, Timothee Mickus, and Erich Round for comments on early versions of this paper.

This work was supported by the Grant No. START/HUM/010 of Grant schemes at Charles University (reg. No. CZ.02.2.69/0.0/0.0/19_073/0016935), as well as a public grant overseen by the French National Research Agency (ANR) as part of the 'Investissements d'Avenir' program (reference: ANR-10-LABX-0083). It has been using data, tools and services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (`https://lindat.cz`), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).

References

Farrell Ackerman and Robert Malouf. 2013. Morphological organization: the low conditional entropy conjecture. *Language*, 89:429–464.

Farrell Ackerman and Gregory T. Stump. 2004. Paradigms and periphrastic expression. In Louisa Sadler and Andrew Spencer, editors, *Projecting Morphology*, pages 111–157. CSLI Publications, Stanford, CA.

Asaf Bachrach and Andrew Nevins, editors. 2008. *Inflectional Identity*. Oxford University Press, Oxford.

Matthew Baerman, Dunstan Brown, and Greville G. Corbett. 2005. *The Syntax-Morphology Interface: A*

Study of Syncretism. Cambridge University Press, Cambridge.

Laurie Bauer. 2004. The function of word-formation and the inflection-derivation distinction. In *Words in their Places. A Festschrift for J. Lachlan Mackenzie*, pages 283–292. Vrije Universiteit, Amsterdam.

James P. Blevins. 2006. Word-based morphology. *Journal of Linguistics*, 42:531–573.

James P. Blevins. 2016. *Word and Paradigm Morphology*. Oxford University Press, Oxford.

Olivier Bonami. 2015. Periphrasis as collocation. *Morphology*, 25:63–110.

Olivier Bonami and Matías Guzmán Naranjo. 2023. Distributional evidence for derivational paradigms. In Sven Kotowski and Ingo Plag, editors, *The Semantics of Derivational Morphology: Theory, Methods, Evidence*, pages 197–235. de Gruyter.

Olivier Bonami and Denis Paperno. 2018. Inflection vs. derivation in a distributional vector space. *Lingue e Linguaggio*, 17:173–195.

Olivier Bonami and Jana Strnadová. 2019. Paradigm structure and predictability in derivational morphology. *Morphology*, 29(2):167–197.

Geert Booij. 1996. Inherent versus contextual inflection and the split morphology hypothesis. In Geert Booij and Jaap van Marle, editors, *Yearbook of Morphology 1995*, pages 1–16. Springer Netherlands, Dordrecht.

Gilles Boyé and Gauvain Schalchli. 2016. The status of paradigms. In Andrew Hippisley and Gregory Stump, editors, *The Cambridge Handbook of Morphology*, pages 206–234. Cambridge University Press.

Andrew Carstairs-McCarthy. 1994. Inflection classes, gender, and the principle of contrast. *Language*, 70:737–788.

Maria Copot, Timothee Mickus, and Olivier Bonami. 2022. Idiosyncratic frequency as a measure of derivation vs. inflection. *Journal of Language Modelling*, 10(2).

Greville G. Corbett. 2010. Canonical derivational morphology. *Word Structure*, 3(2):141–155.

Greville G. Corbett. 2012. *Features*. Cambridge University Press, Cambridge.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530, Osaka, Japan. The COLING 2016 Organizing Committee.

Jerome H Friedman. 2001a. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Jerome H. Friedman. 2001b. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

Jan Hajič, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, and Barbora Štěpánková. 2020. MorfFlex CZ 2.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Roman Jakobson. 1958. Morfologicheskie nabljudenija nad slavjanskim skloneniem (sostav russkix padeznyx form). In *Amer- ican contributions to the fourth international congress of Slavists*. Mouton. Reprinted in English translation in Jakobson (1971).

Roman Jakobson. 1971. *Selected Writings II*. Mouton, The Hague.

Paul Kiparsky. 2005. Blocking and periphrasis in inflectional paradigms. In Geert Booij and Jaap van Marle, editors, *Yearbook of Morphology 2004*, pages 113–135. Springer, Dordrecht.

Michal Křen, Václav Cvrček, Jan Henyš, Milena Hnátková, Tomáš Jelínek, Jan Kocek, Dominika Kováříková, Jan Křivan, Jiří Milička, Vladimír Petkevič, Pavel Procházka, Hana Skoumalová, Jana Šindlerová, and Michal Škrabal. 2021. SYN v9: large corpus of written czech. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Lukáš Kyjánek and Olivier Bonami. 2022. Package of word embeddings of czech from a large corpus. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Marco Marelli and Marco Baroni. 2015. Affixation in semantic space: Modeling morpheme meanings with compositional distributional semantics. *Psychological review*, 122(3):485–515.

Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean. 2000. Boosting algorithms as gradient descent. In *Advances in neural information processing systems*, pages 512–518.

P. H. Matthews. 1991. *Morphology*, 2nd edition. Cambridge University Press, Cambridge.

Timothee Mickus, Olivier Bonami, and Denis Paperno. 2019. Distributional Effects of Gender Contrasts Across Categories. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, volume 2, pages 174–184.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Anna Siewierska. 2004. *Person*. Cambridge University Press, Cambridge.

Andrew Spencer. 2013. *Lexical Relatedness*. Oxford University Press.

Pavol Štekauer. 2015. 14. the delimitation of derivation and inflection. In Peter O. Müller, Ingeborg Ohnheiser, Susan Olsen, and Franz Rainer, editors, *Volume 1 Word-Formation: An International Handbook of the Languages of Europe*, pages 218–235. De Gruyter Mouton.

Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland. Association for Computational Linguistics.

Gregory T. Stump. 1993. On rules of referral. *Language*, 69:449–479.

Gregory T. Stump. 2016. *Inflectional paradigms*. Cambridge University Press, Cambridge.

Gregory T. Stump and Raphael Finkel. 2013. *Morphological Typology: From Word to Paradigm*. Cambridge University Press, Cambridge.

Dieter Wunderlich and Ray Fabri. 1995. Minimalist Morphology: an approach to inflection. *Zeitschrift für Sprachwissenschaft*, 14(2):236–294.

Arnold M. Zwicky. 1985. How to describe inflection. In *Proceedings of Berkeley Linguistics Society 11*, pages 372–386.

The Learnability of the *Wh*-Island Constraint in Dutch by a Long Short-Term Memory Network

Michelle Suijkerbuijk and **Peter de Swart** and **Stefan L. Frank**
Centre for Language Studies, Radboud University
{michelle.suijkerbuijk, stefan.frank, peter.deswart}@ru.nl

Abstract

The current study investigates whether a Long Short-Term Memory (LSTM) network can learn the *wh*-island constraint in Dutch in a way comparable to human native speakers. After establishing with an acceptability judgement task that native speakers demonstrate a clear sensitivity to *wh*-island violations, the LSTM network was tested on the same sentences. Contrary to the results of the native speakers, the network was not able to recognize *wh*-islands and to block gap expectancies within them. This suggests that input and the network's inductive biases alone might not be enough to learn about syntactic island constraints, and that built-in language knowledge or abilities might be necessary.

1 Introduction

In the past decade, artificial neural networks (ANNs) have commonly been used for tasks within the research area of Natural Language Processing, such as machine translation and reading comprehension. This is a remarkable fact for many theoretical linguists, because these networks do not possess the traits considered necessary for language acquisition, such as built-in linguistic knowledge (Chomsky, 1986). Still, recent research has shown that ANNs are able to accurately learn about, for example, number agreement (i.a., Goldberg, 2019; Gulordava et al., 2018), and garden paths (i.a., Frank and Hoeks, 2019; Futrell et al., 2019; van Schijndel and Linzen, 2021). However, not all syntactic phenomena can be learned successfully yet, such as different forms of long-distance dependencies and constraints on these dependencies (Futrell et al., 2019; Wilcox et al., 2022).

One of the first computational investigations on the learnability of long-distance dependencies concerned subject-verb agreement (Gulordava et al., 2018; Linzen et al., 2016). These successful investigations showed that, when Recurrent Neural Networks (RNNs) are presented with the sequence 'The key to the cabinets…', they assign a higher probability to the correct singular verb form 'is' than to the incorrect plural verb form 'are'. Subject-verb agreement is a syntactic phenomenon that frequently occurs in the set of sentences the network is trained on. This makes it easy for the RNN to learn this phenomenon from only the input in combination with its inductive biases, i.e., without any built-in syntactic knowledge necessary. However, to strengthen the claim that RNNs can acquire different long-distance dependencies in this manner, it is important to also investigate dependencies not often seen in the training data set. On the one hand, if these dependencies cannot be learned by the RNN, this suggests that some built-in syntactic knowledge is necessary to learn about these long-distance dependencies. On the other hand, if the RNN can learn these dependencies, it demonstrates that the input and the network's inductive biases suffice, even if the phenomenon itself only infrequently occurs in the input. Island constraints provide an example of such an infrequent long-distance dependency and are central to the current study.

1.1 Island constraints

Filler-gap dependencies are constrained by the type of structure that can contain a gap. Previous research has shown that the filler-gap dependency in (1b) is perceived as unacceptable by most native English speakers in contrast to (1a) (Hofmeister and Sag, 2010).[1]

(1) a. *What*$_i$ did John buy $\__i$?
b. **What*$_i$ do you wonder [$_{\text{wh-phrase}}$ whether John bought $\__i$]?

[1] Gaps are represented by underscores and the *wh*-filler and gap are coindexed with i. Moreover, unacceptability is marked by an asterisk (*).

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 321-331.
Amherst, Massachusetts, June 15-17, 2023

Numerous structures (e.g., the *wh*-phrase in (1b), but also subjects, adjuncts and complex noun phrases) therefore seem to be gap-resistant (Sprouse and Hornstein, 2013; Sprouse et al., 2012). In the literature, these are referred to as *islands* (Ross, 1967), and the unacceptability caused by a filler-gap dependency in an island configuration is called an *island effect*. The current paper will focus on *wh*-islands.

There have been various investigations into the sensitivity of ANNs to the (*wh*-)island constraint, but most, if not all, focused on English. This is a problem because recent literature suggests that recurrent neural networks may have a performance advantage for English-like structural input (e.g., Dyer et al., 2019; Davis and van Schijndel, 2020), while the language learning system must be universal. Therefore, it is important to find out whether these neural networks can successfully learn about grammatical constraints such as islands in other languages as well (e.g., Kobzeva et al., 2023).

The possible performance bias for English-like structural input suggests that performance of the network will be inflated in right-branching languages such as English (i.e., with a basic word order of SVO), but undermined in left-branching and possibly mixed-branching languages (i.e., with a basic word order of SOV; Li et al., 2020).

Dutch employs mixed-branching, which means that a Dutch sentence with a matrix and an embedded clause makes use of two different branching directions; the basic and left-branching word order SOV in the embedded clause and the right-branching word order SVO in the matrix clause (due to V2; Koster, 1975). Crucially, in Dutch, the gap precedes the verb in the embedded clause, as in (2), whereas it follows the verb in English. This difference in word order due to different branching directions makes it interesting to investigate whether neural networks can learn grammatical constraints in Dutch. The current research thus focusses on Dutch as this language is typologically different from English in its word order, but shares many features as well (e.g., morphological complexity).

While there have not yet been any investigations about the performance of neural networks on island constraints in Dutch, there has been some work on the sensitivity of native speakers of Dutch to the *wh*-island constraint. Beljon et al. (2021) showed with an acceptability task that Dutch native speakers are indeed sensitive to the *wh*-island constraint. However, as this is one of only few studies to gather data on islands in Dutch, the current study will try to replicate these findings in a new acceptability judgement task. In addition, to find out whether a neural network performs comparably, a Long Short-Term Memory (LSTM) network is tested on the same sentences the speakers had to judge. The design of the test sentences was largely based on previous computational research examining island constraints in English, which we discuss below.

1.2 Island constraints and neural networks

Different computational investigations have been performed to examine whether neural networks can learn to be sensitive to island constraints. While Chowdhury and Zamparelli (2018) suggest that the networks are affected by processing factors, e.g., the syntactic complexity of islands and the position of this complex structure, Wilcox et al. (2018) argued that LSTMs can correctly learn the syntactic *wh*-, adjunct and complex noun phrase (CNP) island constraints. Wilcox et al. (2019) designed a control study to test whether a processing explanation could explain the results of Wilcox et al. (2018), and showed that LSTMs are able to learn syntactic constraints on filler-gap dependencies instead of simply being sensitive to their complexity. However, they also suggest that the networks are not completely human-like and that they are not able to learn all constraints successfully yet.

Wilcox et al. (2022) decided to combine all the knowledge gathered in these previous studies into the largest investigation to date on the network's learning ability of filler-gap dependencies and island constraints. This investigation used the same experimental design as Wilcox et al. (2018) and the control study introduced by Wilcox et al. (2019) to control for any complexity effects; we used the same design and control in the current research and will discuss them in section 2. Wilcox et al. (2022) showed that *wh*-, adjunct, CNP, left branch, and coordinate structure islands could all successfully be learned by different types of neural networks. Important to note is that these results could not be due to processing factors, as the control study used ruled out this option.

In sum, previous investigations show different results. A general agreement about whether neural networks are able to learn island constraints does thus not exist (yet), and it seems that island constraints are one of the hardest phenomena to

learn for neural networks (Warstadt et al., 2019). This makes it important to investigate why some island constraints (e.g., subject islands) are not successfully learned yet. Moreover, for the island constraints that are already successfully learned in English, it is necessary to investigate whether they can also be successfully learned in other languages. The *wh*-island constraint is, for example, successfully learned in various studies in English (e.g., Wilcox et al., 2022, 2019, 2018), making it interesting to see whether this success is limited to the English language only or whether it can also be achieved in other languages. Therefore, the current research specifically focused on the *wh*-island constraint in Dutch.

2 Methods

To investigate the performance of the native speakers and the LSTM network on the *wh*-island constraint, we constructed experimental and control items that the speakers judged in an acceptability judgement task and that the network assigned surprisal values to.[2] Both the speakers and the network were presented with exactly the same sentences to optimize the comparison.

2.1 Experimental design

The experimental design in the current study was largely based on the interaction design introduced in Wilcox et al. (2018). This interaction design is based on two predictions assumed to be made by the grammar: (1) gaps require fillers, and (2) fillers require gaps. Consequently, the independent variables PRESENCE OF GAP and PRESENCE OF FILLER were crossed, for example in (2) for regular filler-gap dependencies.

(2) Ik weet (**wat/dat**) jij zag dat de bakker
I know (what/that) you saw that the baker
(**koekjes/_**) maakte in de bakkerij.
(cookies/GAP) made in the bakery
'I know (what/that) you saw that the baker made (cookies/_) in the bakery.'

If Dutch speakers indeed assume that fillers require gaps, filled argument positions (*koekjes* 'cookies' in (2)) should be less acceptable and more surprising when a *wh*-filler (*wat* 'what' in (2)) is present. Moreover, if Dutch speakers assume that gaps require fillers, gaps should be less acceptable and more surprising when no *wh*-filler (*dat* 'that' in (2)) is present.

Not only regular filler-gap dependencies were investigated, but also sentences with *wh*-island configurations. Therefore, the factor PRESENCE OF ISLAND was added into the interaction design as well, resulting in the four additional *wh*-island conditions illustrated in (3). The square brackets in (3) indicate the *wh*-island.

(3) Ik weet (**wat/dat**) jij je afvraagt
I know (what/that) you REF wonder
[of de bakker (**koekjes/_**) maakte in
whether the baker (cookies/GAP) made in
de bakkerij].
the bakery
'I know (what/that) you wonder whether the baker made (cookies/_) in the bakery.'

When the gaps and fillers appear in island configurations, the predictions change. First of all, a gap inside an island configuration should never be acceptable and it should be surprising for the network. Second, adding to the predictions made by Wilcox et al. (2018), the presence of a filler will increase the surprisal even more; a gap should not be expected within an island, but coming across a *wh*-filler at the start of the sentence should give rise to the expectation of a gap somewhere else. When this expectation is violated by not encountering a gap somewhere outside of the island, the filler cannot be linked back to a gap, causing the acceptability rating of that sentence to decrease and the surprisal value to increase. This effect should occur in sentences with and without gaps inside the island.

In total, 32 of these experimental item sets were made. The neural network saw all the conditions of each item set (and thus 256 experimental items in total), but each human participant saw only one condition per item set (and thus 32 experimental items in total).

2.2 Control items

As it is argued that humans and neural networks may simply not be able to thread information through syntactically complex constructions (i.e., islands; Keshev and Meltzer-Asscher, 2018; Wilcox et al., 2022, 2019), expectations for gendered pronouns were used to investigate this possibility (similar to the control study designed by Wilcox et al., 2019). To this end, the factors GENDER MATCH and PRESENCE OF ISLAND were

[2]The acceptability judgement task was preregistered. The preregistration can be accessed via https://doi.org/10.17605/OSF.IO/23TEQ

crossed, which resulted in four conditions: a match and mismatch condition for non-islands as in (4a) and for *wh*-islands as in (4b).

(4) a. Ik weet dat de
I know that the
(**meester/juffrouw**) denkt dat
(teacher.MASC/teacher.FEM) thinks that
de leerlingen **hem** begrijpen.
the students him understand
'I know that the (male teacher/female teacher) thinks that the students understand him.'

b. Ik weet dat de
I know that the
(**meester/juffrouw**) zich
(teacher.MASC/teacher.FEM) REF
afvraagt [of de leerlingen **hem**
wonders whether the students him
begrijpen].
understand
'I know that the (male teacher/female teacher) wonders whether the students understand him.'

It is predicted that the sentences in which the semantic gender of the noun phrase (e.g., *meester* (MASC) or *juffrouw* (FEM) 'teacher') matches the gender of the pronoun (*hem* 'him' or *haar* 'her') will be judged as more acceptable and will be less surprising than sentences in which these do not match. However, if there is any trouble in threading information through island configurations, an interaction is expected between GENDER MATCH and PRESENCE OF ISLAND; the gendered expectation effect, i.e., the difference between the sentences with matching and non-matching genders, will be reduced within island configurations. On the other hand, if the native speakers and neural network can work within complex structures, no interaction effect is expected to arise, meaning that the gendered expectation effect will arise in all configurations.

In total, 32 of these control item sets were made. The neural network saw all the conditions of each item set (and thus 128 control items in total), but each human participant saw only one condition per item set (and thus 32 control items in total).

2.3 Filler items

In addition to the experimental and control items, the human participants were also presented with 64 filler items covering the full range of acceptability; 21 acceptable (e.g., regular declarative statements), 22 moderately acceptable (e.g., anglicisms), and 21 unacceptable filler items (e.g., subject-verb agreement errors and word salads). The items and acceptability category (acceptable, moderately acceptable and unacceptable) were based on the filler items used in Beljon et al. (2021) and Kovač and Schoenmakers (2023). The unacceptable filler items were used in the current research to identify participants who appear not to perform the acceptability judgement task faithfully.

2.4 Acceptability judgement task

Participants were presented with 128 sentences (32 experimental, 32 control and 64 filler items) one at a time and were instructed to imagine that these were produced by a native speaker of Dutch that they know well, e.g., a close friend. They were then told to judge these sentences on how good they sound in Dutch (specifically *hoe goed vindt u de zin klinken?* 'how good do you think the sentence sounds?') on a scale ranging from 1 (*Erg slecht* 'very bad') to 7 (*Erg goed* 'very good'), and to base their judgement on their first intuition. Each participant started with 3 filler items to familiarize them with the task. The experiment lasted 15 to 20 minutes and each participant received £3.00.

Ninety-three native speakers of Dutch, recruited from *Prolific*, entered the online experiment in *Qualtrics*. However, 29 were excluded from analyses; 6 because they did not complete the experiment and 23 because they rated more than 2 agreement errors and/or word salads with a rating of 4 or higher on the 7-point scale. The data of the remaining 64 participants ($M_{\text{age}}(SD) = 31.78(9.26)$; range: 20-55; 27 females and 34 males) were analysed.[3]

2.5 The neural network

One LSTM network was trained on a set of sentences extracted from the NLCOW2014 corpus, which comprises individual sentences of Dutch texts collected from the World Wide Web (Schäfer, 2015). Only the first slice, with approximately 37 million sentences, was used in the current research. First, a vocabulary was created by extracting the 20,000 most frequent words of the first slice and adding the set of word types used in the experimental, control and filler items of the current experi-

[3]This specific number of participants, 64, was based on a power analysis performed on unpublished data from a master's thesis. The thesis can be accessed via https://theses.ubn.ru.nl/items/a17d0411-2ed1-49b7-89cc-043540f94e00

ment, if these were not already in the most frequent word list. This resulted in a vocabulary consisting of 20,194 word types. Subsequently, only and all corpus sentences with only words from the vocabulary were selected from the first slice, and these served as training sentences.[4] The total set of training sentences comprised 8,940,314 sentences (144,196,081 tokens).

The LSTM network employed by Frank and Hoeks (2019) was used in the current study without any optimization of the architecture. It was trained on next-word prediction for 5 epochs. First, the words in the vocabulary went through a 300-unit word embedding layer. The word vectors were then passed to a 600-unit recurrent layer and a 300-unit non-recurrent layer. Last, the vectors were passed to the softmax output layer.

To check if the network was well-trained, 2 additional syntactic tests were performed. These tests explored whether the network learned correctly about (a) subject-verb agreement and (b) object-verb order in the embedded clause, a distinctive feature of Dutch (cf. section 1.1). Both are necessary syntactic skills for the network to be able to process a Dutch embedded sentence and any dependencies in it. These tests showed that the network learned both correctly. A more detailed discussion of the items used and the results can be found in Appendix A.

To evaluate the LSTM's performance, the surprisal values were collected that the network assigned to the words in the experimental and control sentences. For the experimental items, surprisal was measured at (a) the verb immediately following the gap or at the filled argument position, e.g., *maakt* 'makes' for sentences with a gap and *koekjes* 'cookies' for sentences without a gap in (2) and (3) (i.e., single-word surprisal values), and (b) summed over all words immediately following the gap or including the filled argument position, e.g., *maakt in de bakkerij* 'makes in the bakery' for sentences with a gap and *koekjes maakt in de bakkerij* 'made cookies in the bakery' for sentences without a gap in (2) and (3) (i.e., summed surprisal values). For the control items, following Wilcox et al. (2019), surprisal was measured summed over the entire sentence, and additionally at the critical pronoun *hem* 'him' or *haar* 'her'.

[4] Sentences with only one word or with more than 50 words, and sentences with a punctuation token that was not a period, comma, exclamation mark or question mark were excluded.

2.6 Data analysis

To compare the performance of Dutch native speakers and the LSTM network, surprisal values are compared to acceptability judgements following the suggestion in Pearl and Sprouse (2015); less probable words and sentences, and thus higher surprisal values, correspond to lower acceptability.

Before the statistical analysis, the raw acceptability judgement scores were converted to z-scores per participant using all items, to correct for individual differences in scale use. Additionally, all independent variables were coded using sum contrast coding, and a box-cox transformation was performed on the standardized judgement scores and the surprisal values so that the transformed data was as close to normally distributed as possible.

For both the analysis of the standardized scores and the (single-word and summed) surprisal values, two linear mixed-effects (LME) models were fitted; one for the experimental items and one for the control items. First, for the experimental items, one LME model was fitted to the standardized scores, one to the summed surprisal values and one to the single-word surprisal values with PRESENCE OF GAP, PRESENCE OF FILLER, PRESENCE OF ISLAND, and their interactions as fixed effects, using the *lmer* function from the *lmerTest* package (Kuznetsova et al., 2017) in R. Second, for the control items, one LME model was fitted to the standardized scores, one to the summed surprisal values and one to the single-word surprisal values with GENDER MATCH, PRESENCE OF ISLAND, and their interaction as fixed effects. The random effect structure for all models was based on the minimal Akaike Information Criterion (AIC). Significance values for the coefficients from all models were calculated using the Satterthwaite approximation in *lmerTest* (Kuznetsova et al., 2017). The interaction effects were further examined using contrasts from the *emmeans* package (Lenth, 2022) in R.

3 Results

3.1 *Wh*-island violations

The final model for the judgements included random intercepts for items and participants. The final model for the single-word surprisal included a random intercept and slope for the interaction between PRESENCE OF GAP and PRESENCE OF FILLER for items, and the final model for the summed surprisal only a random intercept for items.

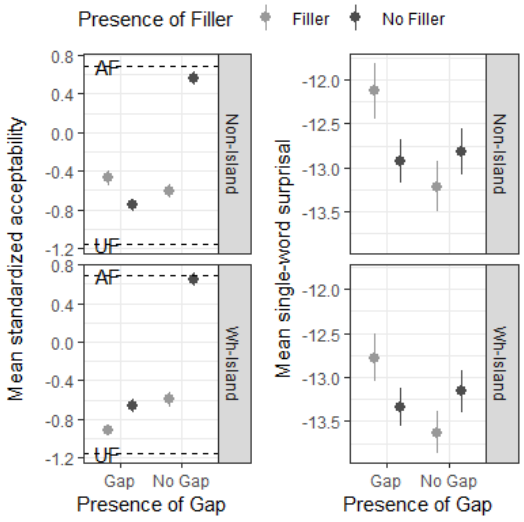


Figure 1: Mean standardized acceptability judgements (left) and mean single-word negative surprisal values (right) for every combination of PRESENCE OF GAP and PRESENCE OF FILLER for non-islands (top) and *wh*-islands (bottom). Dashed lines in the acceptability plot (left) represent the mean acceptability of the acceptable (top line; AF) and unacceptable (bottom line; UF) filler items. Error bars represent standard errors.

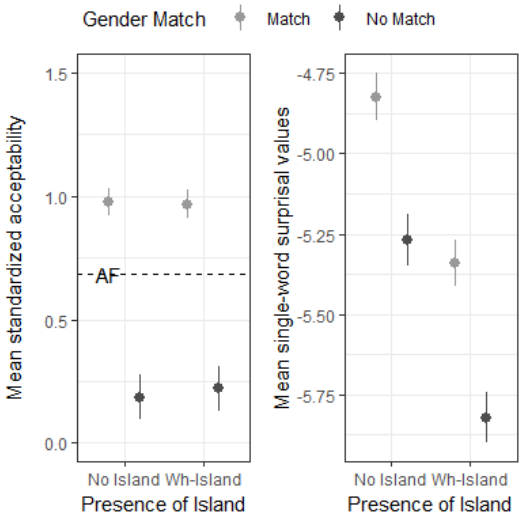


Figure 2: Mean standardized acceptability judgements (left) and mean single-word negative surprisal values (right) in non-islands and *wh*-islands with gender matches and gender mismatches. The dashed line in the acceptability plot (left) represents the mean acceptability of the acceptable filler items (AF). Error bars represent standard errors.

The results of the acceptability judgement task (left) and the LSTM network (right) are shown in Figure 1. On the y-axis of the surprisal plot, the negative surprisal values are used to facilitate the comparison with the judgement plot.

In the acceptability judgement task, a three-way interaction effect was found between PRESENCE OF GAP, PRESENCE OF FILLER, and PRESENCE OF ISLAND ($\beta = -.01$, $SE_\beta = .00$, $p < 001$). For both regular filler-gap dependencies and *wh*-islands, acceptability decreased in sentences with a filled gap when a filler was present ($M_{\text{non-island}}(SD) = -.61(.65)$, $M_{\text{island}}(SD) = -.60(.69)$) as opposed to when it was not ($M_{\text{non-island}}(SD) = .56(.63)$, $M_{\text{island}}(SD) = .65(.62)$) ($p_{\text{non-island}} < .001$, $p_{\text{island}} < .001$). However, the acceptability of regular filler-gap dependencies and *wh*-islands differed when there was a gap. In sentences with a gap, the presence of a filler increased acceptability for regular filler-gap dependencies ($M_{\text{filler}}(SD) = -.47(.70)$, $M_{\text{no filler}}(SD) = -.75(.57)$), but decreased it in a *wh*-island configuration ($M_{\text{filler}}(SD) = -.92(.45)$, $M_{\text{no filler}}(SD) = -.67(.66)$) ($p_{\text{non-island}} < .001$, $p_{\text{island}} < .001$).

For the LSTM network, no three-way interaction effect was found between PRESENCE OF GAP, PRESENCE OF FILLER, and PRESENCE OF ISLAND ($p_{\text{single-word}} = .521$, $p_{\text{summed}} = .634$), but only a two-way interaction between PRESENCE OF GAP and PRESENCE OF FILLER (single-word model: $\beta = -.16$, $SE_\beta = .02$, $p < .001$; summed model: $\beta = -.05$, $SE_\beta = .01$, $p = .002$). This means that the same patterns in surprisal were found for the regular filler-gap dependencies and *wh*-islands.[5] Specifically, surprisal increased in sentences with a filled gap when a filler was present as opposed to when it was not (non-island: $M_{\text{filler}}(SD) = 13.21(2.64)$, $M_{\text{no filler}}(SD) = 12.82(2.46)$; island: $M_{\text{filler}}(SD) = 13.63(2.27)$, $M_{\text{no filler}}(SD) = 13.16(2.21)$) ($p_{\text{non-island}} = .138$, $p_{\text{island}} = .035$), and surprisal decreased in sentences with a gap when a filler was present as opposed to when it was not (non-island: $M_{\text{filler}}(SD) = 12.13(2.96)$, $M_{\text{no filler}}(SD) = 12.92(2.34)$; island: $M_{\text{filler}}(SD) = 12.78(2.53)$, $M_{\text{no filler}}(SD) = 13.34(2.06)$) ($p_{\text{non-island}} < .001$, $p_{\text{island}} = .024$).

3.2 Gendered expectation control

The final model for the judgements included a random intercept and slope for GENDER MATCH for

[5] Only the means and standard deviations of the single-word surprisal are reported as these showed the strongest effects.

items and a random intercept for participants, and the final models for surprisal included a random intercept and slope for PRESENCE OF ISLAND for items.

The results of the participants and the LSTM network on the control items are illustrated in Figure 2. The negative surprisal values were used in the surprisal plot.

For the control items, the native speakers and the LSTM showed the same results. A main effect was found of GENDER MATCH on the standardized acceptability judgements ($\beta = 1.65$, $SE_\beta = .20$, $p < .001$) and on the summed and single-word surprisal values (single-word: $\beta = -.23$, $SE_\beta = .02$, $p < .001$; summed: $\beta = -.05$, $SE_\beta = .02$, $p = .009$), but no interaction effect between GENDER MATCH and PRESENCE OF ISLAND was found on the standardized acceptability judgements ($p = .340$) or the surprisal values ($p_{\text{single-word}} = .597$, $p_{\text{summed}} = .691$). Figure 2 shows that the sentences with a match in gender were more acceptable and less surprising than the sentences with a gender mismatch, and that this effect was the same for non-islands and islands.

4 Discussion

The current research investigated whether an LSTM network showed a similar sensitivity to *wh*-island violations in Dutch as native speakers do. After establishing whether the *wh*-island constraint exists in Dutch in an acceptability judgement task, an LSTM network was tested on the same materials and within the same experimental design to examine whether it showed similar results.

The acceptability judgement task showed that the *wh*-island constraint exists in Dutch, in line with the results by Beljon et al. (2021). Native speakers correctly showed for regular filler-gap dependencies that gaps require fillers and that fillers require gaps, and showed to be sensitive to *wh*-island violations; island configurations were only acceptable without any gaps or fillers present. These findings cannot be explained by islands being too hard to process as the control experiment showed that gender expectations could be maintained within these structures.

The network showed similar results for the regular filler-gap dependencies; it learned that gaps require fillers and that fillers require gaps. Remarkably, however, the same pattern was found within the *wh*-island configuration, contrary to the native speakers; fillers still required gaps, even when that gap then occurs within an island configuration. An LSTM network, trained on nearly 9 million Dutch sentences, does thus not seem to recognize the *wh*-island configuration in Dutch. These findings cannot be explained through processing effects, as the network could maintain gender expectations within island configurations.

While the discrepancy between human judgements and network predictions could be explained by certain design choices of the current research (e.g., the use of judgements and of complex sentences with three sentence-embedding layers), the results could also have been influenced by the architecture of the network, the training procedure, or the word order of Dutch. These factors will be discussed in turn below.

4.1 Acceptability judgements vs. surprisal

While previous research has shown that surprisal is indicative of real-time human language processing (Smith and Levy, 2013), and can thus be compared with human reading times, not much research has compared surprisal values with acceptability judgements yet, giving rise to the concern as to whether this is even possible. Acceptability judgements have been shown to be gradient (see Francis, 2021 for a discussion), which suggests that the knowledge underlying these judgements is probabilistic in nature instead of categorical (Lau et al., 2016). Moreover, multiple previous investigations have argued that acceptability is a concept comparable to probability, as mentioned in section 2.6 (Pearl and Sprouse, 2015; Wilcox et al., 2022). Based on this previous literature, there should be no reason to assume that the judgements and the surprisal values in the current research are not comparable.

4.2 The architecture of the network

The discrepancy between human judgements and network predictions in the current research could be explained by the specific network architecture used. While the current LSTM network does not seem successful in Dutch, other LSTM architectures have been shown to be successful in English; Wilcox et al. (2022) show that two LSTM networks can learn different island constraints successfully in English. The two LSTM networks used were the JRNN as presented in Jozefowicz et al. (2016) and the GRNN as presented in Gulordava et al. (2018). In the JRNN, the input and softmax embeddings are replaced by character convolutional neural net-

works (CNN), making it difficult to compare with the current LSTM. Moreover, the GRNN does not seem comparable either as it differs from the current LSTM in the number of hidden layers. These architectural differences could explain the results obtained for Dutch. For future research, we will thus investigate whether (a) a network more comparable to those used in Wilcox et al. (2022) for English can be successful in Dutch, and (b) the current LSTM would be successful in English.

4.3 Quantity and quality of the training data set

The difference between the human and network's results can also be due to (the size of) the data set the network is trained on. Wilcox et al. (2022) trained the GRNN on 90 million tokens and the JRNN on roughly 1 billion tokens. The current training data set comprised approximately 114 million tokens. The networks used in Wilcox et al. (2022) did not show any qualitative differences in learning success, which seems to suggest that there is no reason to believe that the size of the current data set influenced the network's learning success. While the quantity of the current training data set should thus not be of concern, the quality of the data set could have had an effect.

If the training data sets of the GRNN and the current LSTM are compared, we can identify a difference in syntactic complexity. The GRNN in Wilcox et al. (2022) was trained on English Wikipedia text, while the current training data set comprised sentences extracted from the World Wide Web. It is a well-known fact that Wikipedia text is syntactically quite complex with long and deeply embedded sentences (Yasseri et al., 2012). The current data set seems to have fewer complex sentences as, for example, more coordination conjunction is found in the longer sentences (with more than 45 words) instead of subordinating conjunction. This might mean that the number of complex sentences is smaller in the current data set than in Wikipedia text. This feature could have influenced the network's performance on the experimental items. We followed Wilcox et al. (2022) in the design of the items by using three embedding layers, which might suit Wikipedia text better in syntactic complexity. However, Wikipedia text seems less natural than the current data set, which raises the question to what extent it can be considered natural language input. Future research could use less complex experimental sentences to evaluate the network trained on the current data set, or use a data set more comparable to the one by Wilcox et al. (2022) to train the current model.

Rather than the syntactic complexity of the training data set, it could also be the case that the information in the input (training) data might just not have been good enough to learn about the *wh*-island constraint, as many syntacticians have suggested before (Chomsky, 1965; Pearl and Sprouse, 2013). This could suggest that something else is needed than just external input to learn about the *wh*-island constraint, for example some built-in language knowledge or abilities. While more research is necessary before we can say anything about the need for built-in language knowledge or abilities, our results do suggest that the domain-general learner used in the current study (i.e., the LSTM network trained on nearly 9 million Dutch sentences) is not able to recognize the *wh*-island configuration. Moreover, this domain-general learner has been shown to perform differently than the human speakers, who have been argued to have innate domain-specific knowledge about grammatical constraints (e.g., Chomsky, 1986).

4.4 The Dutch word order

The last factor that could have influenced the results of the current study is word order. The possible performance bias for English-like structural input could mean that performance can be inflated in right-branching languages such as English, but undermined in left-branching and possibly mixed-branching languages such as Dutch (Li et al., 2020). In the current research, combinations of Dutch matrix and embedded clauses were used, and thus a combination of left- and right-branching directions. Crucially, in Dutch, the gap precedes the verb in the embedded clause, which is the other way around in English. This word order difference caused by the difference in branching direction used could have affected the network's results. The current research, however, did not test this hypothesis directly. By replicating the English study by Wilcox et al. (2022), we will be able to compare the network's performance in Dutch and English directly.

In conclusion, in the current research it was shown that an LSTM network does not seem able to recognize the *wh*-island configuration in Dutch and to block gap expectancies within this configuration, unlike native speakers of Dutch. This suggests that input alone might not be enough to learn about island constraints, and that built-in language knowledge or abilities might be necessary. Moreover, it could also suggest that the mixed-branching language Dutch is, in contrast to the right-branching language English, more difficult to grasp for a neural network. Future research is needed to explore the different explanations for the current results.

The data and code can be accessed via `https://doi.org/10.17605/OSF.IO/KT3HE`.

5 Abbreviations

REF	referential pronoun
MASC	masculine
FEM	feminine

References

Maud Beljon, Dennis Joosen, Olaf Koeneman, Bram Ploum, Noëlle Sommer, Peter de Swart, and Veerle Wilms. 2021. The effect of filler complexity and context on the acceptability of wh-island violations in Dutch. *Linguistics in the Netherlands*, 38:4–20.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press.

Noam Chomsky. 1986. *Knowledge of Language: Its Nature, Origin, and Use*. Praeger.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN Simulations of Grammaticality Judgements on Long-Distance Dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144. Association for Computational Linguistics.

Forrest Davis and Marten van Schijndel. 2020. Recurrent Neural Network Language Models Always Learn English-Like Relative Clause Attachment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1979–1990. Association for Computational Linguistics.

Chris Dyer, Gábor Melis, and Phil Blunsom. 2019. A Critical Analysis of Biased Parsers in Unsupervised Parsing. *arXiv preprint*, arXiv:1909.09428.

Elaine J. Francis. 2021. *Gradient Acceptability and Linguistic Theory*. Oxford University Press.

Stefan Frank and John Hoeks. 2019. The interaction between structure and meaning in sentence comprehension: Recurrent neural networks and reading times. In *Proceedings of the Cognitive Science Society*, pages 337–343. Cognitive Science Society.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of NAACL-HLT 2019*, pages 32–42. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. *arXiv preprint*, arXiv:1901.05287.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless Green Recurrent Networks Dream Hierarchically. In *Proceedings of NAACL-HLT 2018*, pages 1195–1205. Association for Computational Linguistics.

Philip Hofmeister and Ivan A. Sag. 2010. Cognitive Constraints and Island Effects. *Language*, 86(2):366–415.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the Limits of Language Modeling. *arXiv preprint*, arXiv:1602.02410.

Mayaan Keshev and Aya Meltzer-Asscher. 2018. A processing-based account of subliminal wh-island effects. *Natural Language and Linguistic Theory*, 37(2):621–657.

Anastasia Kobzeva, Suhas Arehalli, Tal Linzen, and Dave Kush. 2023. Neural Networks Can Learn Patterns of Island-insensitivity in Norwegian. *PsyArXiv preprint*.

Jan Koster. 1975. Dutch as an SOV language. *Linguistic Analysis*, 1:111–136.

Iva Kovač and Gert-Jan Schoenmakers. 2023. An experimental-syntactic take on long passive in Dutch: Unraveling the patterns underlying its (un)acceptability. *Manuscript submitted for publication*, University of Vienna and Radboud University.

Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. 2017. lmerTest Package: Tests in Linear Mixed Effects Models. *Journal of Statistical Software*, 82(13).

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2016. Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge. *Cognitive Science*, 41(5):1202–1241.

Russell V. Lenth. 2022. emmeans: Estimated Marginal Means, aka Least-Squares Means. *R package version 1.8.3*.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2020. On the Branching Bias of Syntax Extracted from Pre-trained Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4473–4478. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 3:521–535.

Lisa Pearl and Jon Sprouse. 2013. Computational models of acquisition for islands. *Experimental Syntax and Island Effects*, pages 109–131.

Lisa Pearl and Jon Sprouse. 2015. Computational Modeling for Language Acquisition: A Tutorial With Syntactic Islands. *Journal of Speech, Language, and Hearing Research*, pages 740–753.

John Robert Ross. 1967. *Infinite Syntax!* Ablex.

Roland Schäfer. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3)*, pages 28–34. Institut für Deutsche Sprache.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Jon Sprouse and Norbert Hornstein. 2013. Experimental syntax and island effects: Toward a comprehensive theory of islands. In *Experimental Syntax and Island Effects*, pages 1–18. Cambridge University Press.

Jon Sprouse, Matt Wagers, and Colin Phillips. 2012. A Test of the Relation Between Working-Memory Capacity and Syntactic Island Effects. *Language*, 88:82–123.

Marten van Schijndel and Tal Linzen. 2021. Single-Stage Prediction Models Do Not Explain the Magnitude of Syntactic Disambiguation Difficulty. *Cognitive Science*, 45(6):1–31.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2019. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *arXiv preprint*, arXiv:1912.00582.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2022. Using Computational Models to Test Syntactic Learnability. *Linguistic Inquiry*, pages 1–44.

Ethan Gotlieb Wilcox, Roger Levy, and Richard Futrell. 2019. What Syntactic Structures Block Dependencies in RNN Language Models? In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*, pages 1199–1205. Cognitive Science Society.

Ethan Gotlieb Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN Language Models Learn about Filler-Gap Dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221. Association for Computational Linguistics.

Taha Yasseri, András Kornai, and János Kertézs. 2012. A Practical Approach to Language Complexity: A Wikipedia Case Study. *PLoS ONE*, 7(11):1–8.

A Appendix

To investigate the performance of the LSTM network on the two additional tests, 15 item sets per phenomenon were created largely based on the item sets used in the main experiment. Each item set consisted of an acceptable and an unacceptable sentence. An example minimal pair for subject-verb agreement can be found in (5) and for object-verb order in (6).

(5) a. Hij weet dat de mevrouw dacht
he knows that the lady thought
dat de jager herten doodt tijdens de
that the hunter deer kills during the
jacht.
hunt
'He knows that the lady thought that the hunter kills deer during the hunt.'

b. *Hij weet dat de mevrouw dacht
he knows that the lady thought
dat de jagers herten doodt tijdens de
that the hunters deer kills during the
jacht.
hunt
*'He knows that the lady thought that the hunters kills deer during the hunt.'

(6) a. Ik weet dat jij denkt dat de bakker
I know that you think that the baker
koekjes maakt in de bakkerij.
cookies makes in the bakery
'I know that you think that the baker makes cookies in the bakery.'

b. *Ik weet dat jij denkt dat de bakker
I know that you think that the baker
maakt koekjes in de bakkerij.
makes cookies in the bakery
*'I know that you think that the baker makes cookies in the bakery.'

First, for subject-verb agreement, it was predicted that the network would assign higher surprisal values to the singular verb (*doodt* 'kills' in (5)) when it followed a plural subject (*jagers* 'hunters' in (5b)) than when it followed a singular subject (*jager* 'hunter' in (5a)). Second, for object-verb order, the network should assign higher surprisal values to the object-verb combination (*koekjes maakt* 'cookies makes' in (6)) when the verb incorrectly precedes the object.

For each phenomenon, an LME model was fitted to the surprisal values with ACCEPTABILITY

as fixed effect using the *lmer* function from the *lmerTest* package (Kuznetsova et al., 2017) in R. The random effect structure for both models was based on the minimal Akaike Information Criterion (AIC). Significance values for the coefficients from the models were calculated using the Satterthwaite approximation in *lmerTest* (Kuznetsova et al., 2017). The final models ultimately included a random intercept for items.

For both phenomena, a main effect of ACCEPTABILITY was found (agreement: $\beta = 1.20$, $SE_\beta = .10$, $p < .001$; order: $\beta = 1.47$, $SE_\beta = .25$, $p < .001$); the acceptable conditions (agreement: $M = 9.22$, $SD = 2.07$; order: $M = 22.57$, $SD = 4.65$) were assigned lower surprisal values than the unacceptable conditions (agreement: $M = 11.61$, $SD = 1.95$; order: $M = 25.52$, $SD = 3.60$).

Towards a Learning-Based Account of Underlying Forms: A Case Study in Turkish

Caleb Belth
University of Michigan
cbelth@umich.edu

Abstract

A traditional concept in phonological theory is that of the underlying form. However, the history of phonology has witnessed a debate about how abstract underlying representations ought to be allowed to be, and a number of arguments have been given that phonology should abandon such representations altogether. In this paper, we consider a learning-based approach to the question. We propose a model that, by default, constructs concrete representations of morphemes. When and only when such concrete representations make it challenging to generalize in the face of the sparse statistical profile of language, our proposed model constructs abstract underlying forms that allow for effective generalization. As a case study, we consider the highly agglutinative language, Turkish. We demonstrate that the underlying forms that our model constructs account for the complexities of Turkish phonology resulting from its multifaceted vowel harmony. Moreover, these underlying forms enable the highly-accurate prediction of novel surface forms, demonstrating the importance of some underlying forms to generalization.

1 Introduction

A traditional conception of phonological theory involves abstract underlying representations (URs) together with phonological processes (stated as rules or constraints) mapping between this abstract level of representation and a concrete, surface-level representation. Debates in the 1960's and 1970's questioned how abstract URs should be allowed to be (Hyman, 2018, p. 597), with a particularly famous article by Kiparsky (1968) arguing that the positing of non-concrete representations should only be done when motivated. Any perception of this debate as fading in subsequent years is probably better attributed to the field moving on to other questions than it is to a satisfactory resolution of the debate (Anderson, 2021).

Indeed, some phonologists have taken the position that URs should not be used in phonological theory because doing so is "(i) wrong, (ii) redundant, (iii) indeterminate, (iv) insufficient, or (v) uninteresting," as Hyman (2018, p. 591) summarized the objections. Meanwhile, much of the work on learning phonology has either focused on surface restrictions (e.g., Hayes and Wilson 2008) or continued to assume URs (e.g., Tesar and Smolensky 1998; Boersma 1997), abstracting away from the question of how (and if) such representations are constructed (see Jarosz 2019 for a summary).

One of the main justifications for the use of underlying representations is to capture generalizations. For example, the form of the English plural affix—[z], [s], or [əz]—depends on the stem-final segment, but is predictable from the stem-final segment, as in (1).

(1) [dɑg-z]
[kæt-s]
[hɔrs-əz]

Positing an underlying /-z/ derived by process into [z], [s], or [əz] allows this generalization to be captured. However such an analysis is not necessary. The allomorphs could each be listed along with a set of sounds each occurs after, or the apparent relationship between singulars and plurals could be ignored altogether and both forms could simply be memorized.[1]

How then are we to choose from these analyses? Is the desire to capture a generalization sufficient motivation to choose the /-z/ analysis? In this work we propose a learning-based approach to this question. Specifically, we propose a computational model that assumes, by default, that underlying forms are fully concrete. The model attempts to form morphological generalizations out of sheer

[1]As one reviewer pointed out, evidence of overgeneralization (e.g., MacWhinney 1978) suggests that memorization is not an empirically-tenable hypothesis in all cases.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 332-342.
Amherst, Massachusetts, June 15-17, 2023

necessity to deal with the sparse statistical profile of language (Yang 2016, ch. 2; Chan 2008).

The question then becomes learning-based: when does surface-alternation of a morpheme prevent the learner from forming morphological generalizations from concrete representations? In some—but critically not all—cases, surface-alternations are pervasive enough to drive the learner to resort to abstract URs in order to effectively generalize. We present the model in § 2.

We evaluate the model on natural-language corpora of the highly agglutinative language Turkish, demonstrating both when abstract URs are necessary for generalization and when they are not (§ 3). When combined with a recent model for learning local and non-local alternations, the proposed model achieves high accuracy generalizing to held-out test words (§ 3.4).

2 Model

2.1 Model Input

The input to the model is a set of morphologically-analyzed surface forms. An example input of nine forms is shown in Tab. 1. These word forms are processed by the model incrementally, modeling the growth of a learner's lexicon.

While morphological segmentation is an important area of study in its own right, we believe it is a justified assumption given experimental evidence that infants can effectively morphologically segment nonce words. These results have been observed for French-learning 11mo-old (Marquis and Shi, 2012) and English-learning 15mo-old (Mintz, 2013) infants. The finding is corroborated by results for 15mo Hungarian-learning infants, despite the high-level of agglutination in Hungarian (Ladányi et al., 2020).

2.2 Model Output

The output of the model is a lexicon, which contains a representation for each morpheme, and a lexicalized list of any input word forms not decomposable into those morphemes. The representation of a morpheme may be concrete or abstract.[2] As discussed by Ettlinger (2008, sec. 4.3.4), a UR can be called *abstract* because it lacks the phonetic detail of an actual speech sound (e.g., /D/ as an alveolar stop lacking a voicing specification), or because it contains different segments from a surface form. For simplicity, we will refer to the representation constructed in the lexicon as a UR, regardless of its abstractness. This assumes, following prior work (§ 4), that each morpheme has a single UR. Future work will consider scenarios where this may not be the case.

Surface Form	Morphological Analysis
1. [buz-lɑr]	'ice-PL'
2. [kɯz-lɑr]	'girl'-PL
3. [el-ler]	'hand-PL'
4. [jer-ler-in]	'place'-PL-GEN
5. [søz-ler]	'word-PL'
6. [dɑl-lɑr-ɯn]	'branch'-PL-GEN
7. [sɑp-lɑr]	'stalk-PL'
8. [jyz-yn]	'face'-GEN
9. [ip-ler-in]	'rope'-PL-GEN

Table 1: An example Turkish input consisting of morphologically-analyzed surface forms.

2.3 Model Description

By default, the model creates a concrete UR for each morpheme. Prior work (§ 4) often resorts to phonological processes to produce the various surface forms of a morpheme at the first instance of surface alternation. Our model differs from this approach by treating underlying forms as concrete even after the first instance of surface alternation. Instead of immediately collapsing surface forms into a single, abstract UR, our model simply lexicalizes all word forms in which a morpheme occurs as something other than its most frequent form. It is only when the resulting lexicalization becomes unsustainable (see § 2.4) that the model then constructs abstract underlying forms from which the surface realizations are derived by morphophonological process.

The pseudocode for the algorithm is shown in (2).[3] As discussed in § 2.1, the input to the model is an incremental stream of morphologically-analyzed surface forms. Whenever the model receives a new surface form (2; step 1), it initially creates a concrete underlying form for each morpheme, storing the most frequent form of the morpheme concretely (2; step 3), and lexicalizes any wordforms that contain a different form of the morpheme (2; step 8). However, if too many word-

[2] We treat surface and underlying representations, whether concrete or abstract, as sequences of segments, where each segment is a set of distinctive features.

[3] Code is available at https://github.com/cbelth/underlying-forms-SCiL

Meaning	UR	Plural Form
PL	/lɑr/	N/A
‘ice’ ‘girl’	/bʊz/ /kʊɹz/	**Stem-PL**
‘hand’	/el/	/el-ler/

Table 2: When the first three words from Tab. 1 enter the lexicon, the stems and plural affix are all stored concretely (left two columns). The plural form of the ‘ice’ and ‘girl’ stems are predictably decomposable into their concrete stems and the PL affix (denoted with the boldface concatenation), so those forms need not be stored in the lexicon. However, with /-lɑr/ as the UR of the plural, the plural form of ‘hand’ cannot be so decomposed, so it is instead lexicalized.

forms in the lexicon are exceptions—where the measurement of “too many” occurs as described in § 2.4—the model instead constructs an abstract UR (2; step 5) and then learns a phonological process, via a separate model (see § 2.6), to account for the resulting alternation.

(2) **Input:** Incremental stream of morphologically analyzed SRs

1. **While** surface form in input **do**
2. – **For** morpheme in segmentation **do**
3. — Morpheme UR ← most freq form
4. — **If** too many alternative forms **do**
5. —— Construct abstract UR
6. —— Learn phonological process
7. — **Else do**
8. —— Lexicalize exceptions

For example, consider the PL suffix after the first 2 (of 9) inputs listed in Tab. 1 have entered the learner’s lexicon. At this point, the model will be storing the only attested surface form [-lɑr] as the concrete UR /-lɑr/.

When the third word enters the lexicon, our model will lexicalize the form ‘hand-PL’ as /el-ler/, rather than immediately constructing an abstract PL morpheme. This is shown in Tab. 2, where each stem and the plural affix have concrete underlying forms, and the plural form of ‘ice’ and ‘girl’ are formed by suffixing the plural to the stem, but the plural form of ‘hand’ is lexicalized.

By the time all 9 words enter the lexicon, however, there will be 4 instances of [-lɑr] and 4 of [-ler], making it no longer sustainable to keep a concrete underlying form. The difference between these two scenarios and, more generally, the decision of when to create an abstract underlying form, is made by the Tolerance Principle (Yang, 2016), as described next.

2.4 When is Abstraction Needed?

In order to detect when the amount of surface alternation that prohibits generalization from concrete representations, the model uses the Tolerance Principle (TP), proposed by Yang (2016). The TP is a cognitively-grounded tipping point, which hypothesizes that children form productive generalizations when the number of exceptions to a proposed generalization results in a real-time processing cost lower than that without the generalization. The exact derivation of the TP is provided by Yang (2016, ch. 3), but rests critically upon the empirical observation of linguistic sparsity. The TP has had much prior success in computational modeling, lexical, and experimental studies (Schuler et al., 2016; Yang, 2016; Richter, 2018; Koulaguina and Shi, 2019; Emond and Shi, 2021; Richter, 2021; Belth et al., 2021; Payne, 2022; Belth, 2023).

Our model’s default treatment of underlying forms as concrete can be stated as a morpheme-specific rule. In the example above, where only the first 2 words of Tab. 1 have entered the lexicon, the rule for the PL form would be (3), which predicts that the PL morpheme is realized as [-lɑr].

(3) If PL then [-lɑr]

The TP threshold, which evaluates a linguistic rule (generalization), is stated in (4), where n is the number of items the rule applies to and e is the number of exceptions to the rule.

(4)
$$e \leq \frac{n}{\ln n}$$

Thus, our model tracks—for each morpheme—the number of observed words in which the morpheme appears (n) and the number of those where surface alternation leads the morpheme to be realized as something other than its hypothesized concrete form (e).

If the (4) threshold is met, then the UR remains concrete and the word forms where the suffix is realized as something else are lexicalized[4] as exceptions. For example, when the 3rd item in Tab. 1

[4]By lexicalization, we mean that the word form is stored in the lexicon verbatim instead of being decomposed into the underlying morphemes. See Tab. 2 for an example.

Morphemes		Word Forms		
Meaning	UR	PL Form	GEN From	PL, GEN Form
PL	/lɑr/	N/A	N/A	N/A
GEN	/in/	N/A	N/A	N/A
‘ice’	/buz/		??	??
‘girl’	/kɯz/	**Stem-PL**	??	??
‘stalk’	/sɑp/		??	??
‘hand’	/el/	/el-ler/	??	??
‘word’	/søz/	/søz-ler/	??	??
‘face’	/jyz/	??	/jyz-yn/	??
‘place’	/jer/	??	??	/jer-ler-in/
‘branch’	/dɑl/	??	??	/dɑl-lɑr-ɯn/
‘rope’	/ip/	??	??	/ip-ler-in/

Table 3: The left two columns contain morphemes—meaning and form (UR); the right three columns contain word forms. Boldface denotes word forms that can be predictably decomposed into concrete underlying forms, while ‘/-/’ notation denotes word forms that must be lexicalized. The ‘??’ denotes word forms that are unknown. Once all nine words from Tab. 1 enter the lexicon, most forms (6 of 9) cannot be predictably decomposed into concrete underlying forms, so the model constructs abstract URs, as described in § 2.5.

enters the lexicon, the realization of PL as [-ler] violates (3). However, with only three word forms containing PL this one exception can be lexicalized, since $1 \leq 3/\ln 3$.

On the other hand, if the (4) threshold is violated—i.e., $n > \frac{n}{\ln n}$—then the model constructs an abstract underlying form. For example, when the 9th item of Tab. 1 enters the lexicon, the realization of PL as [-ler] becomes the 4th of 8 forms in which PL is realized as [-ler] instead of the [-lɑr] predicted by (3). Because $4 > 8/\ln 8$, the model will construct an abstract UR for the PL morpheme.

This is shown in Tab. 3, where the plural is realized as [-lɑr] in 3 plural forms and 1 plural, genitive form, but there are 4 forms that must be lexicalized because they instead have the [-ler] form.[5]

Constructing abstract URs introduces discrepancies between URs and SRs for any word forms containing the morpheme, so our model then passes the (UR, SR) pairs implicit in its lexicon[6] to a model that learns phonological alternations to account for the newly-introduced discrepancies. The process of constructing abstract URs is described in § 2.5 and the process of learning what conditions the alternations is described in § 2.6.

[5]Note that the PL, GEN of ‘branch’ is lexicalized because the GEN affix is realized in a form other than [in], not because of the PL affix, which is why that form does not get counted as an exception in the TP calculation for the PL affix.

[6]See § 2.6 for a description of how the set of (UR, SR) pairs is computed.

2.5 Constructing Abstract URs

The model’s first step in constructing an abstract UR for a morpheme is to create the set of forms that the morpheme is realized as. For example, the forms of the GEN affix attested in Tab. 1 are [-in] / [-ɯn] / [-yn], and of the PL affix are [-lɑr] / [-ler].

Next, the model aligns each of the forms. This is trivial for fixed-length affixes (e.g., the case of the PL affix). If the length of the forms are not all the same, then the model counts the lengths of the morpheme’s realizations. For example, the dative affix can be realized as [-ɑ] or [-e], but may contain an affix-initial [j] when attaching to a morpheme that ends in a vowel. The model thus counts the number of words in which [-ɑ] or [-e] (length 1) is the realization, and the number in which [-jɑ] or [-je] is the realization (length 2), and chooses the most frequent length as the length of the UR. If a shorter length is chosen, the extra segment(s) are treated as epenthesized; if the longer is chosen, they are treated as deleted. For simplicity, we assume that these segments epenthesize or delete on the left, which is a simplification. This process is not guaranteed to generalize to other languages, so future work will develop a more robust alignment process by more tightly combining the problems of abstract UR construction and rule construction.

Once the forms are aligned, the UR is constructed one segment at a time. Each segment is

set to match in features where all realizations of the affix match; features that alternate across forms are unspecified underlyingly. For example, [-lɑr] / [-ler] will lead to /-lAr/, where A is the low, unround vowel with backness unspecified, because both forms agree in the initial and final segments, but the vowel alternates on backness. Similarly, [-in] / [-ɯn] / [-yn] will result in /-Hn/, where H is the high vowel with backness and roundness unspecified, since [i] and [y] differ in backness from [ɯ] while [i] and [ɯ] differ from [y] in roundness.

2.6 Learning Alternations

When the number of words where the morpheme's surface alternation requires the word be lexicalized becomes too great, the model constructs an abstract UR for the morpheme. This abstract UR introduces a discrepancy between the abstract UR and its surface realization. The model thus constructs a set of (UR, SR) pairs from the lexicon, which it passes to a model that learns a phonological process to derive the various surface forms.

For example, when the 9th item from Tab. 1 causes /lɑr/ to no-longer be sustainable as the PL affix UR, the lexicon is as described in Tab. 3. The surface form for the PL forms of the roots 'ice', 'girl', and 'stalk' are computed by concatenating /lɑr/ to the stem (i.e., Stem-PL), and the remaining six known surface forms, which were lexicalized, are extracted directly from the lexicon. Since the PL is being collapsed into /lAr/, each word's UR is computed by replacing the surface realization of the PL affix with this new UR. Thus, the (UR, SR) pairs at this point would be {(/buzlAr/, [buzlɑr]), (/kɯzlAr/, [kɯzlɑr]), (/sɑplAr/, [sɑplɑr]), (/ellAr/, [eller]), (/søzlAr/, [søzler]), (/jyzyn/, [jyzyn]), (/jerlArin/, [jerlerin]), (/dɑllArɯn/, [dɑllɑrɯn]), (/iplArin/, [iplerin])}.

Learning phonological processes from UR-SR pairs is an active area of study, and many models have been proposed (see Jarosz 2019 for an overview). In this work we chose Belth (2023)'s model, which is a cognitively-grounded model that provides a unified ability to learn local and non-local alternations, which is important, given Turkish's non-local vowel harmony combined with local processes like voicing assimilation (see § 3.1).

Belth (2023)'s model is grounded in humans' strong tendency to track adjacent dependencies. For example, artificial language experiments have repeatedly demonstrated that learners more easily learn local phonological processes than non-local ones (Baer-Henney and van de Vijver, 2012) and, when multiple possible phonological generalizations are consistent with exposure data, learners systematically construct the most local generalization (Finley, 2011; White et al., 2018; McMullin and Hansson, 2019).

The Belth (2023) model learns rules to predict the surface form of alternating segments—in this case those that are underlyingly abstract. To do so, the model tracks only dependencies between alternating segments and the segments adjacent to them. If these adjacent segments fail to allow the surface form to be accurately predicted, the model deletes any adjacent segments that prevent the surface form from being predicted, and repeats. The iteratively deleted segments accumulate into a deletion set, the complement of which is interpreted as a tier. The learned rules are applied locally over the tier projection. Because segments are deleted only when adjacent dependencies fail to make the surface form predictable, local processes are a special case, and thus local and non-local processes are learnable by a unified model.

3 Evaluation

This section provides a case study of our proposed model on the highly agglutinative language, Turkish. In § 3.1 we describe some relevant details of Turkish. We then describe the setup of our evaluation in § 3.2. Finally, we present qualitative results in § 3.3 and quantitative results in § 3.4.

3.1 Turkish

Turkish phonology receives attention often because of its apparently complex vowel harmony system. It exhibits both primary front/back harmony and secondary rounding harmony, which is parasitic on height: only [+high] vowels harmonize for roundness. Moreover, Turkish has a number of exceptional suffixes whose vowels do not participate in harmony, and even half-harmonizing suffixes, which have multiple vowels, some of which harmonize and some of which do not. These harmony processes occur alongside other processes, such as local voicing assimilation. The Turkish vowel inventory is shown in (5).

(5)

	front		back	
	unround	round	unround	round
high	i	y	ɯ	u
low	e	ø	ɑ	o

The primary harmony process is observed in affix vowels that alternate between [+back] when the preceding vowel is [+back] and [−back] when it is [−back], as in (6) (examples from Nevins 2010, p. 28; Kabak 2011, p. 3).

(6) [dɑl-lɑr-ɯn] branch-PL-GEN
[jer-ler-in] place-PL-GEN
[ip-ler-in] rope-PL-GEN

The secondary rounding harmony involves suffixal [+high] vowels matching in roundness to the vowel to the left, as in (7) (examples from Nevins 2010, p. 29; Kabak 2011, p. 3). This harmony occurs regardless of whether the vowel to the left is [+high] (7a) or [−high] (7b).

(7) a. [ip-in] rope-GEN
[jyz-yn] face-GEN
[kɯz-ɯn] girl-GEN
[buz-un] ice-GEN
b. [el-in] hand-GEN
[søz-yn] word-GEN
[sɑp-ɯn] stalk-GEN
[jol-un] road-GEN

Some suffixes have vowels that do not participate in harmony. For example, the suffix [-ki] can attach to a temporal or spatial nominal root to yield adjectival forms as in (8), where the suffix surfaces with the vowel [i] regardless of the final vowel of the stem (examples from Oflazer 1994, p. 144). The PL suffix, which alternated in (6), here harmonizes with the [i] vowel (8b), thus surfacing as [e].

(8) a. [øndʒe-ki] '(the one) before'
[jɑrɯn-ki] '(the one) tomorrow'
b. [øndʒe-ki-ler] '(the ones) before'
[jɑrɯn-ki-ler] '(the ones) tomorrow'

The situation gets more complex, as some suffixes are *half harmonizing*, meaning they have two vowels with one harmonizing and one not.[7] An example is shown in (9a), where the first vowel of the abilitative (ABIL) suffix harmonizes with the vowel to the left, but the second vowel is always [−back] [i] even when the first vowel is [+back] (Kornfilt, 2013). The aorist (AOR) suffix vowel then harmonizes with the abilitative's non-harmonizing second vowel [i] in (9a). The example (9b) demonstrates that the AOR suffixal vowel surfacing as [i] in (9a) is indeed due to harmony, as it harmonizes in (9b) with [o].

(9) a. [jɑz-ɑbil-ir] 'write'-ABIL-AOR
[jyz-ebil-ir] 'swim'-ABIL-AOR
b. [ol-ur] 'become'-AOR

Vowel harmony often goes in hand with other phonological processes, such as voicing assimilation. This can be seen, for example, in the locative (LOC) suffix, which exhibits vowel harmony, but begins with an alveolar stop, which assimilates in voicing to the segment to its left, as in (10) (examples from Dobrovolsky 1982; Çöltekin 2010; Kornfilt 2013).

(10) [byro-dɑ] 'office'-LOC
[ev-de] 'house'-LOC
[ʤep-te] 'pocket'-LOC

In the remaining subsections, we demonstrate how our proposed model elegantly accounts for these complexities in Turkish (§ 3.3), and how this allows for novel surface forms to be accurately predicted (§ 3.4). First, though, we introduce the setup and data we used for our experiments (§ 3.2).

3.2 Setup and Data

To simulate learning in Turkish, we constructed two Turkish datasets consisting of frequency-annotated and morphologically-analyzed surface forms (see below). To simulate one learning trajectory, we sampled words with replacement from the corpus, weighted by frequency. Each time a new word form is sampled, the learner adds it to its lexicon. We then investigate the underlying forms of each morpheme, seeing which are concrete and which are abstract (§ 3.3). We then evaluate how accurately the model, combined with a model for learning alternation rules, allows novel surface forms to be predicted (§ 3.4).

We constructed two datasets, called MorphoChallenge and CHILDES. The first used data from MorphoChallenge (Kurimo et al., 2010), which contains a large Turkish corpus annotated

[7] The term *half harmonizing* is from Nevins (2010), but one reviewer pointed out that, in principle, other fractions of vowels (1 of 3) could harmonize.

with word frequencies. To generate morphological analyses of words, we used Çöltekin (2010, 2014)'s finite state morphological analyzer, which is designed for Turkish. This is similar to the process used in the MorphoChallenge, but is publicly available.[8] We dropped any word in MorphoChallenge that had fewer than 25 occurrences or for which the morphological analyzer failed to provide an analysis. We also removed forms with affixes that are analyzed by Çöltekin (2010, 2014) as having multiple underlying forms. For example, the highly irregular aorist suffix is sometimes described as having four underlying forms: /-Ar/, /-Hr, /-z/, /-null/. Future work will consider scenarios where multiple URs are necessary. This resulted in 22,315 frequency-annotated and morphologically-analyzed surface forms, which we transcribed into IPA.

The second dataset is derived from the child-directed speech in the Aksu (Slobin, 1982) and Altinkamis corpuses of the CHILDES database (MacWhinney, 2000). We computed the frequency of each word in the corpuses and used the same process as above to morphologically analyze each word. This dataset is much smaller, so we did not exclude words with low corpus counts from this dataset. The resulted in 1,727 frequency-annotated and morphologically-analyzed surface forms, transcribed into IPA.

Note that some Turkish suffixes exhibit deletion/epenthesis to avoid CC or VV clusters. These additional processes are at present ignored, because the implementation provided by Belth (2023) was designed for harmony and disharmony. Future work will extend the implementation to epenthesis and deletion by incorporating Belth (In Press)'s model, which handles such processes.

3.3 Suffixes: Abstract and Concrete

Remarkably, the apparent complexity of Turkish vowel harmony, discussed in § 3.1, vanishes when we investigate the output of our model.[9] As before, we will let A denote the Turkish low, unround vowel with backness unspecified (extensionally, {e, ɑ}) and H be the Turkish high vowel with both backness and height unspecified (extensionally, {i, y, ɯ, u}). Moreover, we will use D to denote the alveolar stop with voicing unspecified (extensionally, {d, t}).

[8] https://github.com/coltekin/TRmorph

[9] This analysis is performed on a random, frequency-weighted 80% sample of the MorphoChallenge dataset.

We will walk through the complexities exemplified by (6)-(10) one-by-one. First, the PL suffix in (6), which has a low unrounded vowel, participates in front/back harmony, but not rounding harmony because it is not a [+high] vowel. Our model constructed the underlying form /-lAr/ for this suffix, capturing the fact that it only harmonizes for backness.

The GEN suffix in (6)-(7) has a [+high] vowel and participates in both primary and secondary harmony. Our model constructed the underlying form /-Hn/ for this suffix, which captures the surface alternation of this morpheme.

Next, the [-ki] suffix in (8) does not participate in harmony, and our model consistently represents it with a concrete form /-ki/.

For the abilitative suffix in (9), our model abstracts the first, harmonizing vowel, but keeps the second, non-harmonizing vowel concrete -/Abil/.

Lastly, the UR for the locative suffix in (10) is constructed with both segments abstract /-DA/, capturing both the voicing assimilation of the initial alveolar stop and the vowel harmony of the second segment.

These underlying forms allow Belth (2023)'s model to learn two rules, which allow for the accurate prediction of novel surface forms. On the resulting (UR, SR) pairs, Belth (2023) learns a vowel harmony rule, which targets both /A/ and /H/ vowels, and enforces harmony with respect to their unspecified values: [back] for /A/ and both [back] and [round] for /H/. The model automatically constructs a vowel tier and enforces harmony locally over that tier (see Belth 2023 for details). Belth (2023)'s model also learns a local voice assimilation rule, which causes /D/ to take its [voice] value from the segment to its left.

It is worth noting that others—in particular Nevins (2010)—have similarly argued that Turkish vowel harmony can be elegantly accounted for with an underspecification approach. Our model builds on Nevins (2010)'s observations by providing an explicit computational model that constructs underlying forms, which turn out to be consistent with this analysis.

As a further analysis, we show the 10 most frequent affixes in a 1K word sample of the CHILDES corpus in Tab. 4, along with the UR that our model constructed for each. Of the 10 affixes, 7 have been collapsed into abstract forms. However, there are 3 forms (P1S, lH, P2S) that were quite frequent,

Affix	UR	Abstract
PL	/-lAr/	Y
P3S	/-H/	Y
P1S	/-m/	N
GEN	/-Hn/	Y
DAT	/-A/	Y
ACC	/-H/	Y
LOC	/-DA/	Y
VN:INF	/-mA/	Y
lH	/-lɯ/	N
P2S	/-n/	N

Table 4: Top 10 most frequent affixes in a random, frequency-weighted sample of 1K words from the CHILDES dataset, and the URs that our model learned. See http://coltekin.net/cagri/trmorph/trmorph-manual.pdf for a description of affix names.

but are still able to be stored concretely. The P1S and P2S affixes do not have alternating segments in Turkish, so it is expected that these would be concrete. The "lH" affix, as captured by its name, can surface with any high vowel. However, in the training data, the [-lɯ] form occurs 25 out of 32 times, so the 7 words where it surfaces as something else are lexicalized ($7 <= 32/\ln 32$).

3.4 Quantitative Evaluation

We also evaluated how the model enables generalization, when paired with a model for learning phonological alternations. We used our model in tandem with Belth (2023)'s model to learn to map a stem and morphological analysis of a surface form to an actual surface form. For example, given the stem [dɑl] and morphological analysis Stem-PL-GEN, our model's underlying forms for -PL and -GEN are concatenated to the stem to form a UR, to which the generalizations learned by Belth (2023) can then be applied to predict a surface form, such as [dɑllarɯn].

We ran the model on both datasets, simulating incremental learning by sampling words with replacement and weighted by frequency, and adding them to the lexicon when sampled. As this process incrementally adds words to the lexicon, our model operates as described in (2). In 250-word increments (i.e., every time the lexicon grows by 250 unique words), we evaluated the model by using the rules learned by Belth (2022)'s model—on our learned underlying forms—to predict the surface form of all the words not in the lexicon. We carried out 5 simulations on each dataset, using different random seeds for sampling on each.

The results are shown in Fig. 1, where the x-axis shows the incremental growth of the learner's lexicon (i.e., the training size), and the y-axis shows the accuracy at predicting novel surface forms at that point during training. The accuracy is computed over all surface forms not currently in the training data. Each subfigure is for one of the two datasets. The MorphoChallenge results (Fig. 1a) are reported up to a size of 3K words, so the test results are on 10s of thousands of novel words.

The model's performance appears to be consistent with acquisition studies. Altan (2009) found that Turkish-speaking children as young as 2;0 extend vowel harmony to nonce words. Studies across languages reveal that a child's vocabulary is quite modest at this age, with an upper bound around 1K words (Fenson et al., 1994; Hart and Risley, 1995; Szagun et al., 2006; Bornstein et al., 2004). The model's performance on both datasets is above 90% accuracy when its vocabulary contains 1K words.

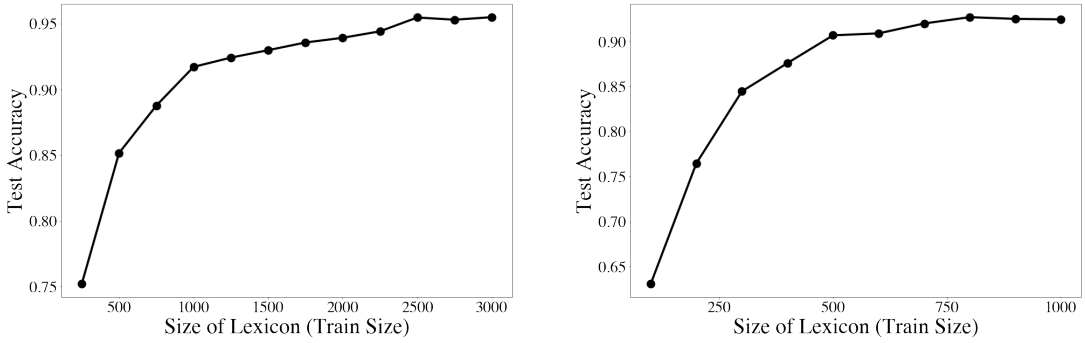
3.4.1 Error Analysis

Of the errors, around 52% result from the model having a concrete form of an affix, which it then errantly predicts for a novel word that exhibits alternation in that affix. For example, there are insufficient forms in the training data to make /ɯp/ as the concrete CV:IP affix prohibitive ($e = 5 <= n = 13/\ln 13$), even though vowel harmony leads it to sometimes surfaces with other high vowels. As a result, novel words like [gel-ip], which take the [ip] form of the affix are mispredicted.

About 47% of the errors are the result of vowel harmony or consonant assimilation being predicted for a novel form that exceptionally does not involve harmony. For example, the word [sɑɑt-ler] 'watch-PL' is predicted by our model to be [sɑɑt-lar] because the UR for the plural suffix is /lAr/, as it systematically harmonizes. According to a Wiktionary search,[10] the root [sɑɑt] is of Arabic origins. Because Arabic has a different vowel system, vowels in Arabic loan words may conform to the Turkish vowel system when entering Turkish, and thus sometimes behave oddly. Indeed, Altan (2009) observed that children may overextend vowel harmony to such words.

The remaining 1% of errors result from very low

[10] https://en.wiktionary.org/wiki/saat#Turkish



(a) MorphChallenge

(b) CHILDES

Figure 1: Our proposed model's accuracy (averaged over 5 simulations) at predicting novel surface forms. The x-axis shows the growth of the learner's lexicon (i.e., the training size).

frequency affixes which are simply unattested in the training data.

4 Prior Work

Tesar (2014) and Hua et al. (2020) focus on theoretical analyses of the nature of the problem of learning URs. O'Hara (2017); Rasin et al. (2018); Ellis et al. (2022) proposed computational models, but evaluate on small, phonology-textbook-like data, not large, natural-language corpora.

Cotterell et al. (2015) also predominately models textbook-like problems, but presents some limited analysis on more realistic corpora. However, these corpora only involve very simple morphological paradigms involving a single suffix, and present to the model a fairly curated subset of the corpus that isolates the relevant morphophonological process.

Richter (2021) studies the question of when allophonic surface segments are collapsed into an abstract underlying segment, focusing on the English flap [ɾ] allophone of /T/. While Richter (2021) focuses on allophones, our proposed model is inspired by it and can be viewed as extending the same principles to morphophonological alternations.

Of these prior models, we were only able to get access to code for Cotterell et al. (2015) and Rasin et al. (2018), which we were unable to get to run on our large datasets. In future versions of this work, we intend to implement some of these existing models in order to compare their performance and behavior to that of our proposed model.

5 Conclusion

This work proposed a learning-based account of underlying forms, taking the highly agglutinating language of Turkish as a case study. The proposed model starts with concrete underlying representations and constructs abstract URs only in cases where doing so helps to form generalizations that deal with the sparsity of morphological forms in the learner's input.

The model constructs abstract underlying forms when they are critical for generalization, but allows for concrete forms when abstraction is unecessary. This flexibility is at the core of the model's success, as evidenced by the fact that the representations of Turkish suffixes in § 3.3 are minimally abstract. For example, the half-harmonizing suffixes consist of concrete segments except for the single, harmonizing vowel. Similarly, exceptional, non-harmonizing suffixes remain fully concrete.

When combined with a model for learning local and non-local alternations, the proposed model achieves >95% accuracy predicting the surface form of held-out test words.

This work presents a preliminary case study in Turkish. Future work will evaluate the model on other languages. Moreover, the algorithm takes as input morphologically-segmented surface forms. As discussed in § 2.1, there is experimental evidence that children are able to perform morphological segmentation. In future work, we will attempt to bring the problems together, jointly segmenting surface forms, learning underlying forms, and morphophonological grammars.

References

Asli Altan. 2009. Acquisition of vowel harmony in Turkish. *Dilbilim 35. Yıl Yazıları*, pages 9–26.

Stephen R. Anderson. 2021. *Phonology in the Twentieth Century*. Number 5 in History and Philosophy of the Language Sciences. Language Science Press, Berlin.

Dinah Baer-Henney and Ruben van de Vijver. 2012. On the role of substance, locality, and amount of exposure in the acquisition of morphophonemic alternations. *Laboratory Phonology*, 3(2):221–249.

Caleb Belth. 2022. A learning-based account of phonological tiers. *Linguistic Inquiry*. Under Review.

Caleb Belth. 2023. Learning non-local phonological alternations via automatic creation of tiers. In *Oral Presentation at the 97th Annual Meeting of the LSA*.

Caleb Belth. In Press. A learning-based account of local phonological processes. *Phonology*.

Caleb Belth, Sarah Payne, Deniz Beser, Jordan Kodner, and Charles Yang. 2021. The greedy and recursive search for morphological productivity. In *CogSci*.

Paul Boersma. 1997. How we learn variation, optionality, and probability. In *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, volume 21, pages 43–58. Citeseer.

Marc H Bornstein, Linda R Cote, Sharone Maital, Kathleen Painter, Sung-Yun Park, Liliana Pascual, Marie-Germaine Pêcheux, Josette Ruel, Paola Venuti, and Andre Vyt. 2004. Cross-linguistic analysis of vocabulary in young children: Spanish, dutch, french, hebrew, italian, korean, and american english. *Child development*, 75(4):1115–1139.

Erwin Chan. 2008. *Structures and distributions in morphology learning*. Ph.D. thesis, University of Pennsylvania.

Çagri Çöltekin. 2010. A freely available morphological analyzer for turkish. In *LREC*, volume 2, pages 19–28.

Çagri Çöltekin. 2014. A set of open source tools for turkish natural language processing. In *LREC*, pages 1079–1086.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3(0):433–447.

Michael Dobrovolsky. 1982. Some thoughts on turkish voicing assimilation. In *Calgary Working Papers in Linguistics*, volume 7, pages 1–6.

Kevin Ellis, Adam Albright, Armando Solar-Lezama, Joshua B Tenenbaum, and Timothy J O'Donnell. 2022. Synthesizing theories of human language with bayesian program induction. *Nature communications*, 13(1):5024.

Emeryse Emond and Rushen Shi. 2021. Infants' rule generalization is governed by the Tolerance Principle. In *Proceedings of the 45nd annual Boston University Conference on Language Development*, pages 191–204.

Marc Ettlinger. 2008. *Input-driven opacity*. University of California, Berkeley.

Larry Fenson, Philip S Dale, J Steven Reznick, Elizabeth Bates, Donna J Thal, Stephen J Pethick, Michael Tomasello, Carolyn B Mervis, and Joan Stiles. 1994. Variability in early communicative development. *Monographs of the Society for Research in Child Development*, pages i–185.

Sara Finley. 2011. The privileged status of locality in consonant harmony. *Journal of memory and language*, 65(1):74–83.

Betty Hart and Todd R Risley. 1995. *Meaningful differences in the everyday experience of young American children.* Paul H Brookes Publishing, Baltimore, MD.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Wenyue Hua, Adam Jardine, and Huteng Dai. 2020. Learning underlying representations and input-strictly-local functions. In *Proceedings of the 37th West Coast Conference on Formal Linguistics*.

Larry M Hyman. 2018. Why underlying representations? *Journal of Linguistics*, 54(3):591–610.

Gaja Jarosz. 2019. Computational modeling of phonological learning. *Annual Review of Linguistics*.

Bariş Kabak. 2011. Turkish vowel harmony. *The Blackwell companion to phonology*, pages 1–24.

Paul Kiparsky. 1968. *How abstract is phonology?* Indiana University Linguistics Club.

Jaklin Kornfilt. 2013. *Turkish*. Routledge.

Elena Koulaguina and Rushen Shi. 2019. Rule generalization from inconsistent input in early infancy. *Language Acquisition*, 26(4):416–435.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge 2005-2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95. ACL.

Enikő Ladányi, Ágnes M Kovács, and Judit Gervain. 2020. How 15-month-old infants process morphologically complex forms in an agglutinative language? *Infancy*, 25(2):190–204.

Brian MacWhinney. 1978. *The acquisition of morphophonology*. Monographs of the Society for Research in Child Development. University of Chicago Press.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for analyzing talk. transcription format and programs*, volume 1. Psychology Press.

Alexandra Marquis and Rushen Shi. 2012. Initial morphological learning in preverbal infants. *Cognition*, 122(1):61–66.

Kevin McMullin and Gunnar Ólafur Hansson. 2019. Inductive learning of locality relations in segmental phonology. *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, 10(1).

Toben H Mintz. 2013. The segmentation of sublexical morphemes in english-learning 15-month-olds. *Frontiers in psychology*, 4:24.

Andrew Nevins. 2010. *Locality in vowel harmony*, volume 55. Mit Press.

Kemal Oflazer. 1994. Two-level description of turkish morphology. *Literary and linguistic computing*, 9(2):137–148.

Charlie O'Hara. 2017. How abstract is more abstract? learning abstract underlying representations. *Phonology*, 34(2):325–345.

S Payne. 2022. *When collisions are a good thing: the acquisition of morphological marking*. Bachelor's thesis, University of Pennsylvania.

Ezer Rasin, Iddo Berger, Nur Lan, and Roni Katzir. 2018. Learning phonological optionality and opacity from distributional evidence. In *Proceedings of NELS*, volume 48, pages 269–282.

Caitlin Richter. 2018. Learning allophones: What input is necessary. In *Proceedings of the 42nd annual Boston University Conference on Language Development.* Cascadilla Press.

Caitlin Richter. 2021. *Alternation-Sensitive Phoneme Learning: Implications For Children's Development And Language Change*. Ph.D. thesis, University of Pennsylvania.

Kathryn D Schuler, Charles Yang, and Elissa L Newport. 2016. Testing the tolerance principle: Children form productive rules when it is more computationally efficient to do so. In *CogSci*, volume 38, pages 2321–2326.

Dan I Slobin. 1982. Universal and particular in the acquisition of language. *Language acquisition: The state of the art*, 57.

Gisela Szagun, Claudia Steinbrink, Melanie Franik, and Barbara Stumper. 2006. Development of vocabulary and grammar in young german-speaking children assessed with a german language development inventory. *First Language*, 26(3):259–280.

Bruce Tesar. 2014. *Output-driven phonology: Theory and learning*. 139. Cambridge University Press.

Bruce Tesar and Paul Smolensky. 1998. Learnability in optimality theory. *Linguistic Inquiry*, 29(2):229–268.

James White, René Kager, Tal Linzen, Giorgos Markopoulos, Alexander Martin, Andrew Nevins, Sharon Peperkamp, Krisztina Polgárdi, Nina Topintzi, and Ruben van De Vijver. 2018. Preference for locality is affected by the prefix/suffix asymmetry: Evidence from artificial language learning. In *NELS*, pages 207–220.

Charles Yang. 2016. *The price of linguistic productivity: How children learn to break the rules of language*. MIT press.

Unbounded recursion in two dimensions, where syntax and prosody meet

Edward P. Stabler
University of California, Los Angeles
stabler@ucla.edu

Kristine M. Yu
University of Massachusetts Amherst
krisyu@linguist.umass.edu

Abstract

Both syntax and prosody seem to require structures with unbounded branching, something that is not immediately provided by multiple context free grammars or other equivalently expressive formalisms. That extension is easy, and does not disrupt an appealing model of prosody/syntax interaction. Rather than computing prosodic and syntactic structures independently and then selecting optimally corresponding pairs, prosodic structures can be computed directly from the syntax, eliminating alignment issues and the need for bracket-insertion or other ad hoc devices. To illustrate, a simple model of prosodically-defined Irish pronoun displacement is briefly compared to previous proposals.

Since phonological structures do not show a principled bound on length, those structures must allow unbounded branching or unbounded depth or both. There is significant controversy about how the balance is struck (Selkirk, 1996, 2011; Ito and Mester, 2012). Idsardi (2018) suggests that the issue can be largely set aside if the appearance of phonological structure derives entirely from the syntax, with a transduction that concatenates segmental material and inserts 'boundary symbols'. But Yu (2021) points out that boundary symbol insertion should not be accidental, stipulated; if there are no prosodic constituents, then we need another explanation of 'boundary' distribution. Rigorous studies of these matters are often based on grammars and automata that do not provide mechanisms for unbounded branching. This absence may obscure part of our picture of the syntax-prosody interface.

For syntax, Chomsky (1961, 1963, 2018) observes that standard rewrite grammars do not provide unbounded branching:

> The failure of strong generative capacity of [phrase structure grammar] . . . is a failure of principle, as shown by unstructured coordination: e.g., "the man was old, tired, tall,. . . , but friendly". Even unrestricted rewriting systems fail to provide such structures, which would require an infinite number of rules. The more serious failure, however, is in terms of explanatory adequacy. (Chomsky, 2018, p.132)

Chomsky's remarks about this are discussed in Lasnik (2011) and Lasnik and Uriagereka (2022, pp.15-20). Lasnik (2011) notes that Chomsky and Miller (1963, p. 298) actually consider this context free rule for adjective coordination:

$$\text{Predicate} \rightarrow \text{Adj}^n \text{ and Adj} \qquad (n \geq 1).$$

However, as Lasnik notes:

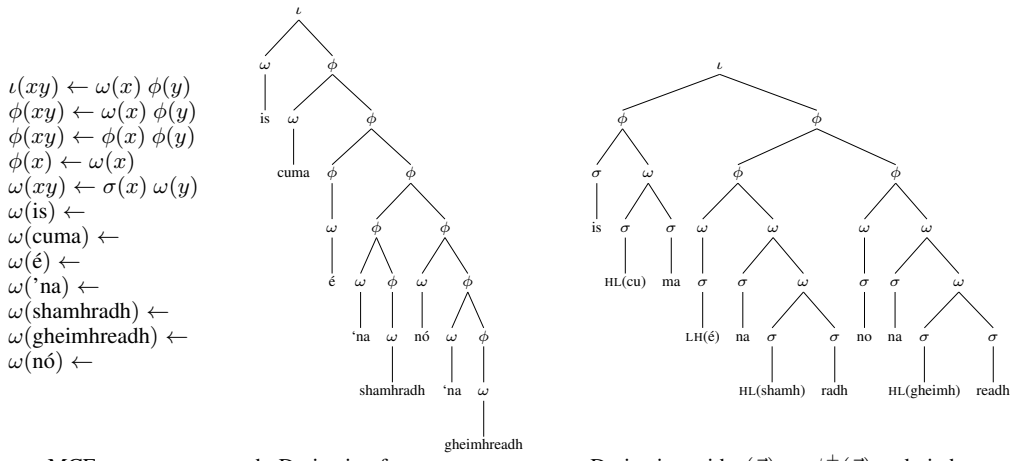
> Chomsky and Miller indicate that there are "many difficulties involved in formulating this notion so that descriptive adequacy can be maintained. . . ". But they do not elaborate on this point. It would surely be interesting to explore this. . . (Lasnik, 2011, p.361)

That option is explored here.

Inspired by Kleene (1956), unbounded branching can be added to phrase structure rewrite grammars by allowing the Kleene star $*$ on the right side of any rule.[1] Yu (2022), reviewed in §1, proposes that prosodic constituency and dependencies can be specified by multi bottom up tree transducers or, equivalently, multiple context free grammars. These can also be extended with $*$ on the right side of any rule, accommodating unbounded prosodic branching. In recent syntax too, the evidence supports unbounded branching. Neeleman et al. (2023) defends unbounded branching for coordination, and briefly reviews the long history of such proposals. McInnerney (2022b) argues for unbounded branching in adjunction. And Chomsky (2021, p.20) recently proposes a $*$-extension of merge, in his rule 'D'.

[1] This idea is used in finite state toolkits (Beesley and Karttunen, 2003; Hulden, 2009; Gorman and Sproat, 2021), and $*$-extended context free grammars are commonly used to define programming languages (Wirth, 1977; Albert et al., 2001; Martens and Niehren, 2005; Jim and Mandelbaum, 2010; Borsotti et al., 2023). Pattis (1994) argues that context free grammars with unbounded branching should be taught on the first day of your first class in Computer Science.

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 343-356.
Amherst, Massachusetts, June 15-17, 2023



a. MCF grammar b. Derivation from grammar a c. Derivation with $\iota(\vec{x}) \leftarrow \phi^+(\vec{x})$ and pitch accents

Figure 1: $*$-MCF prosody for (1)

A $*$-extension of minimalist grammar is proposed in §2, providing an analog of Chomsky's rule D. And since the primary causal influences in prosody and syntax differ, it is natural to define them separately. But this creates a puzzle about how the respective influences interact in linguistic performance. A syntax-prosody interface inspired by Bennett et al. (2016) is proposed, one that allows prosodically conditioned pronoun postposing of weak pronouns in Irish. These pronouns can appear middle of a coordinate structure, suggesting a non-syntactic placement. With a syntax for Irish coordination that allows unbounded branching, postposing of these weak elements can occur in the generation of prosodic structure.

1 $*$-Prosodic structure

Yu (2021) points out that many phonologists argue for structures with branching constituencies that finite state automata do not provide. And Yu (2022) observes that certain multiple dependencies, sometimes marked with arcs in representations of phonological structure (Pierrehumbert and Beckman, 1988), can be captured and made explicit in 'finite state multi bottom up tree transducers' (MBOTs) or, equivalently, in 'multiple context free grammars' (MCFGs). A simple MCFG is presented in Figure 1a using the logic-based notation of Kanazawa et al. (2011). Each rule is a conditional, with the back arrow ← pronounced "if", and with variables over strings on the right that get concatenated on the left side of each rule. The first rule Figure 1a says "xy is an ι if x is a ω and y is a ϕ". The last says "the string *nó* is an ω".

Here, we also extend MCFGs with the Kleene star and plus. Any category C on the right side of a rule can be starred, C^*, meaning that it may occur 0 or more times, where the strings of this sequence are adjacent in the sequence $\vec{x}$. We also allow C^+ which is the same as C followed by C^*. So the rule

$$\iota(\vec{x}) \leftarrow \phi^+(\vec{x})$$

says "the strings of $\vec{x}$, concatenated, are an ι if they are the strings of one or more occurrences of ϕ". An instance of that rule is applied at the root of Figure 1c. And the rule from Chomsky, mentioned in the introduction, is:

$$\text{Predicate}(\vec{x} \text{ and } y) \rightarrow \text{Adj}^+(\vec{x})\ \text{Adj}(y).$$

That says "$\vec{x}$, concatenated with *and*, concatenated with string y, is a Predicate if $\vec{x}$ are the strings of 1 or more Adj, and y is also an Adj". We will also write LH(é) to indicate that é has the pitch accent LH, and similarly for HL. These extensions do not change MCFG expressive power (Appendix A). MCFGs are 'multi' in allowing categories to classify multiple strings – relevant in §3.

Example. Consider the Irish (1) from Bennett et al. (2016), in which we added syntactic bracketing for the coordinate structure:

(1) is cuma [é ['na shamhradh] [nó ['na gheimhreadh]]]
COP.PRES no.matter it PRED summer or PRED winter
'It doesn't matter if it's summer or winter'

Assuming the syntactic structure in Figure 4 (excluding the conjunct *'na fhómar*), the prosodic structure expected following the syntax-prosody mapping principles of Match Theory (Selkirk,

2011; Elfner, 2012) is shown in Figure 1b. In brief, optimality-theoretic MATCH constraints enforce that clausal projections correspond to intonational phrases (ι), maximal projections to phonological phrases (ϕ), and heads to prosodic words (ω). However, Bennett et al. (2016, (104)) propose that the prosodic structure in fact phrases pronoun é together with the first conjunct *'na shamhradh* in a single ϕ, like in Figure 1c.

Briefly, to explain this, they propose that prosodic markedness constraints are ranked above MATCH constraints, following Elfner (2012, §4.2). The key prosodic markedness constraints are: (i) EQUALSISTERS (Bennett et al., 2016, (48)), which assigns a violation when sisters are not of the same prosodic category (Myrberg, 2013), (ii) STRONGSTART (Bennett et al., 2016, (55)), which penalizes ϕ- and ι-phrases with leftmost daughters that are "prosodically dependent", i.e., syllables (σ), and (iii) BINARITY, which penalizes nodes that are not binary branching. Here we assume that BINARITY is applicable only to ϕ-nodes, following Elfner (2012, §4.2), and that EQUALSISTERS is applicable only to nodes above the prosodic word (since Myrberg (2013) and Bennett et al. (2016) consider only above the level of the prosodic word). In addition, Bennett et al. (2016, p. 198) assume that a prosodic word must contain a stressed syllable, which we can encode as an inviolable CULMINATIVITY constraint.

While the tree in Figure 1b incurs no MATCH constraint violations, it incurs five EQUALSISTERS violations due to $\langle\omega, \phi\rangle$ daughter pairs, as well as three BINARITY violations due to unary branches to *é*, *shamhradh*, and *gheimhreadh*; moreover, *is* and *'na* (but crucially, not *é*) are stressless clitics and thus incur violations of CULMINATIVITY. In contrast, the prosodic tree in Figure 1c incurs a number of MATCH violations, but no BINARITY violations and only single STRONGSTART and EQUALSISTER violations due to the phrasing of the daughters *is* and *cuma*. The structure in Figure 1c with pronoun *é* linearized preceding the conjuncts is only optimal when *é* occurs in its strong, stressed form. When *é* occurs in its weak, unstressed form, it cannot form a prosodic word on its own—only a syllable. If the ω node over *é* in Figure 1b was deleted, leaving just a σ, violations of EQUALSISTERS and STRONGSTART would be incurred.

2 ∗-Minimalist grammar

Minimalist grammars (MGs) are weakly equivalent and closely related to MCFGs (Harkema, 2001a; Michaelis, 2001) and can be similarly extended to unbounded branching, leaving weak expressive power unchanged (Appendix A). Here we adapt the version of MG in Kobele (2021), which has only positive and negative feature occurrences, where expressions are formed by merging expressions in which each negative occurrence is 'mated' with a positive occurrence.

We use only one polarity relation following Kobele (2021) and others.[2] Initially, let a minimalist grammar (MG) be a finite set of lexical items that associate phonological forms with feature-based formulas as follows:

feature	::=	V | D | A | C | wh | . . .
		| feature$^+$ |feature* | X
non-empty-conj	::=	feature | feature . non-empty-conj
conj	::=	ϵ | non-empty-conj
formula	::=	conj $\multimap$ non-empty-conj
lexical-item	::=	phonological-form : formula

In any formula, features in the antecedent conjunction on the left are are negative; those in consequents positive. When an antecedent is empty, instead of $\epsilon \multimap$ a.b or $\multimap$ a.b, we often write a.b.

∗-Merge. We extend the usual definition of binary merge to to allow any number of constituents to be combined in one step:

$$\mathrm{M}(A, B, C_1, \ldots, C_n) = \{A, B, C_1 \ldots, C_n\}.$$

At least 2 constituents are required, so it is sometimes convenient to write $A, B, \vec{C}$ for $A, B, C_1, \ldots, C_n$ $(n \geq 0)$. Sets are unordered, of course, but order would be redundant since, as will become clear, heads and subcategorized elements are distinguishable by their labels.

Labels. Derivations begin with *numerations*, which are defined here as finite sequences of lexical and derived elements. Merge applies to numeration elements, replacing them. And the merge steps of a successful derivation produce complexes which can be assigned a label by function ℓ. A lexical or derived structure A whose first unmated feature is

[2] MGs often use 2 canceling pairs (=x selects x, and +x licenses -x), but here we use 1. A head (negative occurrence of x) 'mates' or 'cancels' a non-head (positive occurrence of x). Eliminating the move/merge distinction arguably makes scope reconstruction less surprising (Sportiche, 2017; Chomsky, 1995, §3.5). Cf. CMGs (Stabler, 2011), e-MGs (Chesi, 2021), and Horn linear logic (Kanovich, 2015).

Labels are defined with 3 cases (lexical items, internal merge, and external merge, respectively):

$$\ell(A) = \begin{cases} A : \{\alpha \multimap \beta\} & \text{if } A \text{ is a lexical item } w : \alpha \multimap \beta \\ A : \gamma & \text{if } A = \{B, C, \vec{D}\},\ C : F \in \ell(B), \gamma = m(\ell(B), \{C : F\}) \text{ is defined, and } \&(\ell(C), \vec{D}) \\ A : \gamma & \text{otherwise, if } A = \{B, C, \vec{D}\}, \gamma = m(\ell(B), \ell(C)) \text{ is defined, and } \&(\ell(C), \vec{D}) \end{cases}$$

Tentatively, $\&(\alpha, \vec{D})$ iff every element of $\vec{D}$ has label α.
And the 'mating' function calculates the labels of complexes, for the third case of ℓ:

$$m(S[f.\alpha \multimap \beta], T[B : \{f.\gamma\}]) = \begin{cases} \{\alpha \multimap \beta\} \cup S \cup T & \text{if } \gamma = \epsilon \text{ and smc}(S \cup T) \\ \{\alpha \multimap \beta, B : \gamma\} \cup S \cup T & \text{if } \gamma \neq \epsilon \text{ and smc}(\{B : \gamma\} \cup S \cup T), \end{cases}$$

where $X[\alpha]$ is a set X containing formula α and then X is the result of removing that element, and where $\text{smc}(X)$ iff no two formulas in X have the same first unmated feature.

Figure 2: MG label checking

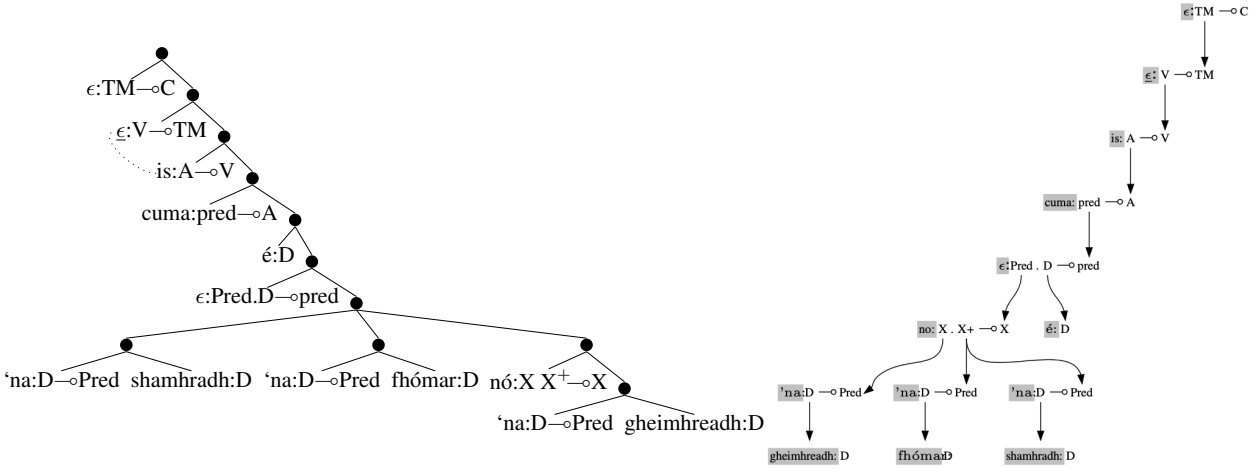


Figure 3: Left, structure for (2): a set in which leaves are lexical items, internal nodes are sets, arcs are $\in$ relations. A dotted arc is added to indicate PF head movement, independent of syntax-derived set. Right, the corresponding dependency tree (with feature-checking arcs).

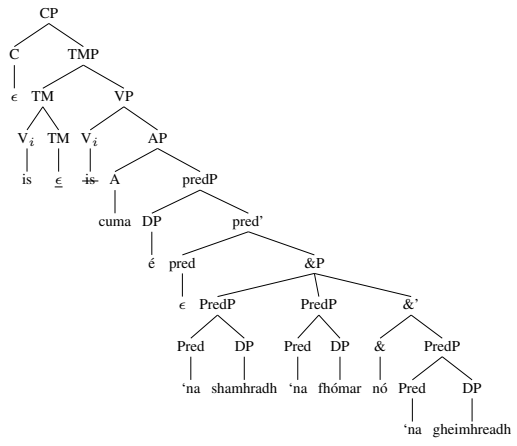


Figure 4: X-bar tree for derivation of Figure 3.

negative f can *mate* with a lexical or derived structure B whose first unmated feature is positive f. Labeling maps a lexical item or derived set A to a pair $A : F$, where F is the set that contains the formula of the head, but with mated feature removed, together with the pairs $\vec{B} : \vec{G}$ of subconstituents $\vec{B}$ with unmated positive features $\vec{G}$, as detailed in Figure 2.

As in previous MGs, ℓ requires embedded positive elements to satisfy the 'shortest move constraint' (smc): $\ell(A, B)$ is undefined if A, B have any first positive feature in common. The mating m then applies to the labels. Writing $N[A, B, \vec{C}]$ when A and $\vec{C}$ are in numeration N and either (i) $\vec{B} \in N$ or (ii) $B \in \ell(A)$, let $N[\text{M}(A, B, \vec{C})]$ be the result of letting $\{A, B, \vec{C}\}$ replace A, B and $\vec{C}$

in N.[3] We call steps using labeling condition (i) *external merge* and steps using (ii) *internal merge* or *move*. Note that a move-over-merge condition is imposed in the definition of the labeling ℓ in Figure 2 – it's the last, 'otherwise' case.[4]

The labeling of pairs A, B is extended to the labels of $A, B, \vec{C}$ by requiring each element of C to have the same label as B, and assigning the complex the same label it would have if $\vec{C}$ were empty. In lexical entries, f^+ is a special feature that allows labeling of $\vec{C}$, with 1 or more elements with first negative feature f. For convenience, in any lexical item, we also allow variable X to be instantiated with any single feature.

Derivations. Now a rule R, building syntactic structures from elements of a numeration, can be formulated like this:

$$\frac{N[A,B,\vec{C}]}{N[\mathrm{M}(A,B,\vec{C})]}\ \text{(R) if } \ell(\mathrm{M}(A,B,\vec{C})) \text{ is defined.}$$

A structure is *complete* when it has exactly one unmated feature, that feature is on its head, and it is positive. And a derivation from numeration is complete when we have derived a single complete structure. The grammar defines the set of complete structures derived from numerations of its elements. For any feature c, let L_c be the set of sets of non-empty phonological forms at the leaves of completed structures with that feature.

Linearization. Unlike rule R, parsers construct derivations from numerations of zero or more non-empty and often ambiguous phonological forms, and linear order matters. For any grammar G define

$$G(x) = \begin{cases} \{A \in G |\ A = x : F\} & \text{if } x \text{ is phonological} \\ \{x\} & \text{otherwise.} \end{cases}$$

Tentatively, let's adopt the Kayne-like idea that first-mated elements are pronounced to the right of the head later-mated elements on the left, with elements pronounced only in their derivationally latest positions.[5]

[3] Appendix C has a complete implementation of R. With compilers that avoid 'destructive' operations, 'replacement' of $A, B, \vec{C}$ by $\{A, B, \vec{C}\}$ need involve no deletion, but rather a change in how the elements are accessed (Wadler, 1992).

[4] Following Kobele (2021). Sometimes merge-over-move is assumed (Epstein et al., 2012; Chomsky, 2000, p.106), but that has been challenged on empirical grounds (Shima, 2000; Castillo et al., 2009; Abels, 2012, §4.3.1). Careful discussion of the these alternatives, and their interaction with the smc and island constraints, is beyond the scope of this brief study.

[5] See e.g. Kayne (2020, 1994); Collins and Kayne (2020); Johnson (2017); Biberauer et al. (2014); Nunes (1999).

Order is further complicated by 'head movement', which we assume is non-syntactic, morphologically-driven (Harizanov and Gribanova, 2019; Chomsky, 2021, i.a.). A morphological feature of a selecting head can attract the head of a selected complement to its left.

Let's call this rule K:

$$\frac{N[x,y,\vec{z}]}{N[\mathrm{M}(A,B,\vec{C})]}\ \text{(K)}\quad \text{if}\quad \begin{array}{l} A \in G(x), B \in G(y), \vec{C} \in G(\vec{z}), \\ \ell(\mathrm{M}(A,B,\vec{C})) \text{ is defined, and} \\ \text{if this is } B\text{'s last mating, then} \\ \quad (\text{ if this is } A\text{'s first mating,} \\ \quad\ \text{then } A, B, \vec{C} \text{ are adjacent in } N; \\ \quad\ \text{else, } \vec{C}, B, A \text{ are adjacent }), \text{ and} \\ \text{a morphological feature of } \underline{\text{A}} \text{ can attract} \\ \text{the phonetic head of first merged B.} \end{array}$$

A simple model of rule K is implemented by the minimalist grammar mechanisms of Stabler (2001) and Stanojević (2019).[6]

In the long tradition of generalizations about linear precedence, this idea is among the simplest.[7] MGs adopting this idea are very expressive, defining a mildly context sensitive class of languages (Michaelis, 2001; Harkema, 2001b).

Example, continued. Consider this 3-coordinate elaboration of the previous example:

(2) is cuma é 'na shamhradh, 'na
COP.PRES no.matter it PRED summer, PRED
fhómhar nó 'na gheimhreadh
autumn, or PRED winter
'It doesn't matter if it's summer, autumn or winter'

We assume that the head movement shifts the copula from V to a tense-modality position TM below the complementizer C (McCloskey, 2022). And we assume that a predP small clause is the complement of the adjective. Then a structure similar to the one proposed by Bennett et al. can be defined by this lexicon, indicating the morphological feature of the empty head-raising TM by underlining it:

ϵ: TM $-\!\circ$ C	$\underline{\epsilon}$: V $-\!\circ$ TM	is: A $-\!\circ$ V
cuma: pred $-\!\circ$ A	ϵ:Pred.D $-\!\circ$ pred	
'na: D $-\!\circ$ Pred	nó: X $X^+ -\!\circ$ X	
shamhradh: D	fhómar: D	gheimhreah: D

[6] A further extension is proposed for coordinate structures by Torr and Stabler (2016): when all coordinates have the same head, they can all be 'adjacent' to the selecting head in the sense required for head movement in (K). And note that Figure 2's requirement that coordinates have identical types is too strong. Relaxing that condition to handle ellipsis, etc., the higher order structures of type logics are valuable (Kubota and Levine, 2021, and references cited there). Even in that powerful system, it is not yet clear how to avoid lexical redundancies and other issues (Morrill and Valentín, 2017). Kobele (2019) extends a minimalist grammar with similarly higher-order structures, but further exploration of these issues is left for future work.

[7] Cf. e.g. Shieber (1984); Daniels and Meurers (2004); Abels and Neeleman (2012); Cinque (2017); Kusmer (2020); Stanojević and Steedman (2021); Roberts (2021).

From any numeration that contains exactly 1 occurrence of each of these elements, we can derive the complete structure depicted by Figure 3 left, where internal nodes are sets with downward arcs to their respective elements. Figure 3 also shows the corresponding dependency graph, and Figure 4 an X-bar structure.[8] Clearly, with numerations of elements from this 10 element lexicon, we can derive not only (2) but also (1) and an infinite number of other structures of category C, with any number of coordinates.

3 The meeting point

Bennett et al. (2016) note that there are variants of (1) in which pronoun é is prosodically weak and postposed, with prosodic structures shown in Figure 5: [9]

(3) is cuma 'na shamhradh é nó 'na
COP.PRES no.matter PRED summer it or PRED
gheimhreadh
winter

(4) is cuma 'na shamhradh nó 'na
COP.PRES no.matter PRED summer or PRED
gheimhreadh é
winter it

For a syntactician, (3) is a puzzle. Why and how could a pronoun be displaced into the middle of a coordinate structure? Bennett et al. suggest that this happens for reasons that were already needed in the account of (1). Because the pronoun é is prosodically weak, it doesn't adjoin at the left edge of the first conjunct in (1) like in Figure 1c, where it would incur both StrongStart and EqualSister violations. Instead, it avoids violating StrongStart via postposing. In fact, the Bennett et al. OT account of (1) extends almost immediately to (3) and (4) once we allow the prosody to consider candidates with displacement. Here we show that proposal has a transparent and efficient computational implementation.

A common idea is that the relation Gen pairs each syntactic structure input with all possible prosodic trees, or all prosodic trees that yield the same string of pronounced elements. Then Match can require that each syntactic XP correspond to a ϕ in the prosodic structure. But the number of possible trees can be very large, and how are corresponding (XP,ϕ) pairs found? Counting each XP and requiring a corresponding number of ϕ is unnecessarily nonlocal and inefficent. Requiring that each XP have an ϕ dominating the same words is worse – many XPs can have the same words, so how do we keep track of them?

A natural idea is to represent the set of candidates for any input with a finite state transducer. A tree transducer is simply a device that traverses an input tree, going into one of finitely many states at each point. Bottom-up transducers traverse the input from the leaves up to the root. Traversing the input, the output tree is extended in each step by rules that depend on the current state and the next symbol of the input tree. A transducer that is 'multi' has states that can have several output subtrees at once, allowing it to move things up through the tree, to be assembled into the structure later. We also allow our transducers to be 'extended', which means that a rule can look at more than just one symbol of the input at a time, allowing simpler rules. So we use XMBOTs, finite state extended multi bottom up tree transducers (Engelfriet et al., 2009).

In a transduction from an input to an output tree, an alignment is established by the operation of the transduction itself. Traversing an input XP, the transducer will either output the corresponding ϕ or not, and the latter case can be penalized. And more generally, when all the constraints are themselves definable by finite state transducers, an important result from string-based OT carries over to the setting: a guarantee that optimal structures can be computed efficiently (Ellison, 1994; Eisner, 1997; Albro, 1997; Heinz et al., 2009).[10] In this setting, instead of considering each candidate one-by-one, we apply constraints to the finite state grammar that generates all the candidates. Large candidate sets are then unproblematic, so we can allow candidates with displacement, and candidates that skip levels in the prosodic hierarchy.[11]

[8] Standard sets related by membership are multidominance structures, but they are simpler than some multidominance structures of earlier proposals (Gärtner, 2002, 2014; Citko, 2011). MG dependency graphs are used by Kobele (2021), Salvati (2011), Stabler (1999), inspired by proof nets (Moot and Retoré, 2012; Moot, 2002; Girard, 1987). And for computing X-bar structure, see e.g. Stabler (2013, App.B).

[9] Cf. Chung and McCloskey (1987); McCloskey (1999); Duffield (1995); Adger (1997, 2007); Mulkern (2003, 2009); Elfner (2012); Bennett et al. (2016); Windsor et al. (2018); Kusmer (2020).

[10] See Daland (2014) and Heinz and Idsardi (2017) for brief comparison of this computational model with others prominent in phonology.

[11] This tree-based strategy, expressing Gen and constraints with composable finite state transducers, was suggested by Graf (2012a,b), and is the natural option here. In contrast, Kalivoda (2018, (179)), Bellik and Kalivoda (2017, Appendix) and Kalivoda and Bellik (2020, §4) define Gen as a set of

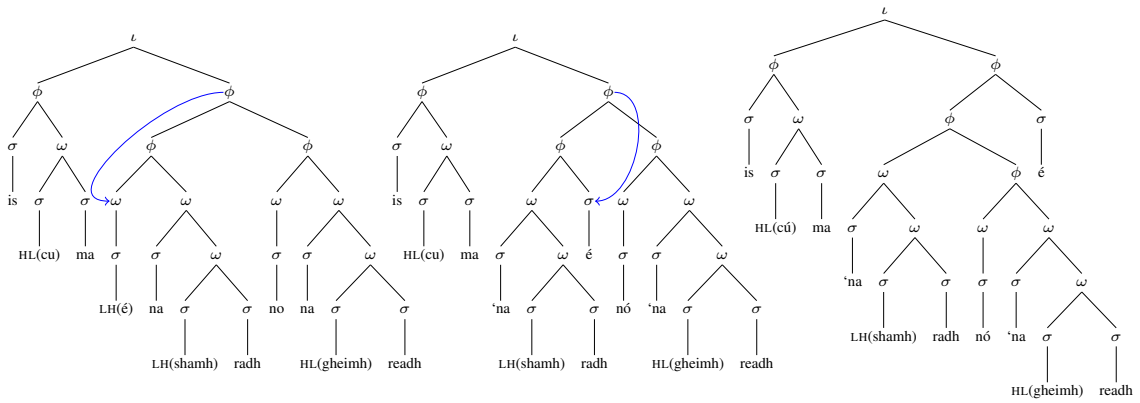
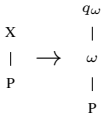


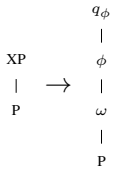
Figure 5: Prosody for (1), (3), (4): attaching é left of sister's 1st daughter, right of that daughter, and right of sister

Engelfriet et al. (2009) point out that MCFGs are just MBOTs that compute string yields, and so the $*$-extension of XMBOTs is similar. And for any two linear XMBOTs, we can construct a single XMBOT that computes their composition. When GEN is an XMBOT, and each constraint is an XMBOT that marks some structures every time the constraint is violated, then we can compose GEN with the top-ranked constraint for an XMBOT that still generates all candidates but with additional marks on the steps that violate constraints. Then, using Dijkstra's algorithm, paths that produce more constraint violations than necessary can be pruned to generate only structures that are optimal with respect to that first constraint. Iterating this step to apply constraints from the most highly ranked to the lowest, pruning suboptimal paths in each result, the algorithm stops when there is only one remaining candidate or when all constraints have been evaluated. This exactly simulates a tableau evaluation, and is guaranteed to be efficient even when the candidate sets are large or infinite.[12]

For illustration, let's take a few steps in the derivation of a prosodic structure, beginning with the familiar X-bar structure in Figure 4, except, as in §1, we leave out indices and the middle coordinate. For this example, we use 4 states $q_\omega, q_\phi, q_\iota, q_\epsilon$, with q_ι the final state. For nonempty head category X (that is, for V,A,Pred,&) with phonetic content P, we have the rule:



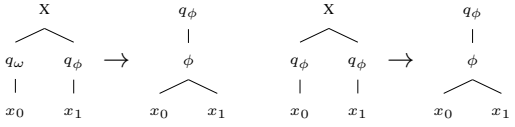
For phrasal category XP with phonetic content P:



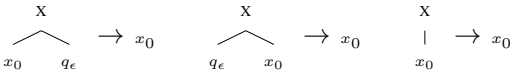
For any category X:

That set of rules, applied bottom up, replaces all the terminal elements of Figure 4 by states with subtrees.

For internal nodes, variables x_0, x_1 range over subtrees. For non-head categories X,



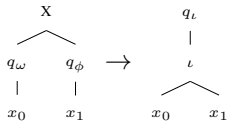
And for any category X



pairs. They require that the order of pronounced elements in the input and output are the same, so prosodic displacements are not among the candidates. Bellik et al. (2021, fn3) clarifies that their trees also do not include level-skipping, apparently disallowing e.g. ϕ parents of σ in Figure 1b,c. Kusmer (2020, §6.1) defines GEN to allow the (much larger) set of pairs in which all orders of pronounced elements appear among the output candidates, and does not confront the computational problem. Dolatian et al. (2021) does propose using a transducer to map from syntax to prosody, but does not use OT.

[12] Frank and Satta (1998) credit Paul Smolensky with noting that this kind of approach, with a pruning step that does not require any finite bound on violations, can be non-finite-state, unlike e.g. 'lenient composition' (Karttunen, 1998). A referee conjectures that our constraints are 'global' (Jäger, 2002), guaranteeing finite-stateness. And other regular versions of OT might extend naturally to prosodic trees, e.g. Lamont (2022). We leave these broader issues for later work.

Finally, we add a 9th rule:



These rules suffice to map the X-bar tree to the prosodic structure in Figure 1a, along with many other candidate structures. (See Appendix C.)

To guarantee closure under composition, note that these rules are linear in the sense that each variable on the left appears at most once on the right. And note that the rules are nondeterministic, because the left side of the last rule – a rule for ι – is identical to the left side of one of the rules for ϕ. Among the properties of these rules that are linguistically important: phonetically empty structure is discarded; and MATCH-governed alignments are completely transparent. That is, rules that process heads but do not introduce an ω are violating, as are rules processing XP without introducing a ϕ, and rules that process clauses without introducing ι. And of course we can track alignments in more complex rule sets where the alignments are not quite so transparent.

Figure 1b is good for MATCH, but violates other constraints that may be ranked more highly, like BINARITY. We can easily see which rules create non-binary structures. So if, for a given input, it is possible to avoid those rules, we can throw them out – the algorithm informally described above automates the discovery of such non-optimal offenders. More importantly, XMBOTs, because they are 'multi', can move also things around. That is, in effect, they can delay the construction of the ϕ dominating the conjuncts in the structures of Figure 5 until the pronoun comes into view. This allows the more optimal, displaced alternatives in the middle and right trees of Figure 5 to be constructed when é is weak, since these alternatives are available.

All the constraints mentioned in the §1 sketch of the Bennett et al. (2016) proposal can be defined as XMBOTs. So efficient computation of optimal prosody from $*$-MG derivations is guaranteed.[13]

[13]Dolatian et al. (2021) points out that the stress rule proposed for coordinate structures by Wagner (2010) is not computed by any XMBOT. The empirical basis of Wagner's proposal could be challenged, or, as Dolation et al. speculate, Wagner's stress rule could be implemented by allowing a very restricted copying. We leave this for future work.

4 Parsing and future work

Seki et al. (1991) present an MCFG parsing algorithm that is succinctly reviewed by Kallmeyer (2010, §7.1), who says "The idea is that once all the predicates in the right side of a rule have been found, we can complete a left side". To allow star and plus categories C^*, C^+ on the right side, there are two cases. Non-empty categories are expand as possible in the chart, exactly as if there were rules with any number of Cs. Empty categories, on the other hand, can introduce cycles in the chart of completed constituents, just as right recursion over empty categories does.

$*$-MGs with Rule K can also be parsed directly. In the bottom-up MG parsing of Harkema (2001b, §4.4), for example, the required adjustment is almost identical to the one for Seki's MCFG parser. Instead of arbitrarily many MCFG rules, Harkema has merge, treated in 5 cases, but the complete rules are essentially the same. So for starred features in a merge rule, any number of constituents is allowed to match. An implementation is linked in fn. 17.

For any MG structure, we compute optimal prosodic structure by $*$-extended transductions, with 'unranked' trees. There are already tree transducer libraries (Bahr, 2012; May and Knight, 2006; Genet and Tong, 2001; Rival and Goubault-Larrecq, 2001), but an up-to-date tree-based toolkit designed specifically for linguists would be useful, analogous to the finite state string toolkits mentioned in fn. 1. This would provide an efficient way to explore a large range of proposals about syntax/phonology interaction, even in cases where large or infinite candidate sets need to be assessed.

Looking at unbounded coordination in Irish also raises linguistic issues that are left for future work. Consulting Irish linguists, it seems, at least to some, that the pronoun in the 3 coordinate case can be initial or final, but nowhere inside the coordinate structure.[14] It seems unlikely that BINARITY should hold in this and longer, list-like coordinations.

More generally, it is not clear that this is the right way for syntax to meet prosody, but the formal model perhaps makes some aspects of the situation clearer. And the $*$-extension of MG syntax should be unified with previous ideas about 'persistent' features (Stabler, 2011; Graf and Kostyszyn, 2021), and with the broader TSL program (Heinz et al., 2011; Graf, 2022).

[14]We are grateful for advice, judgements and references from James McCloskey, Dónall Ó Baoill, and Ryan Bennett.

References

Klaus Abels. 2012. *Phases: An Essay on Cyclicity in Syntax*. Linguistische Arbeiten.

Klaus Abels and Ad Neeleman. 2012. Linear asymmetries and the LCA. *Syntax*, 12(1):25–74.

David Adger. 1997. VSO order and weak pronouns in Goidelic Celtic. *Canadian Journal of Linguistics*, 42(1-2):9–29.

David Adger. 2007. Pronouns postpose at PF. *Linguistic Inquiry*, 38(2):343–349.

Jürgen Albert, Dora Giammarresi, and Derick Wood. 2001. Normal form algorithms for extended context-free grammars. *Theoretical Computer Science*, 267(1–2):35–47.

Daniel M. Albro. 1997. Evaluation, implementation, and extension of primitive optimality theory. Master's thesis, UCLA.

Patrick Bahr. 2012. Modular tree automata. In *Mathematics of Program Construction*, pages 263–299. Springer LNCS 7432. Code: https://hackage.haskell.org/package/compdata-automata.

Kenneth R. Beesley and Laurie Karttunen. 2003. *Finite State Morphology*. CSLI.

Jennifer Bellik, Junko Ito, Nicholas Kalivoda, and Armin Mester. 2021. Matching and alignment. In Haruo Kubozono, Junko Ito, and Armin Mester, editors, *Prosody and Prosodic Interfaces*. Oxford University Press.

Jennifer Bellik and Nicholas Kalivoda. 2017. Syntax-prosody in optimality theory. Technical report, University of Califoria, Santa Cruz.

Ryan Bennett, Emily Elfner, and James McCloskey. 2016. Lightest to the right: An apparently anomalous displacement in Irish. *Linguistic Inquiry*, 47(2):169–234.

Theresa Biberauer, Anders Holmberg, and Ian Roberts. 2014. A syntactic universal and its consequences. *Linguistic Inquiry*, 45(2):169–225.

Angelo Borsotti, Luca Breveglieri, Stefano Crespi Reghizzi, and Angelo Morzenti. 2023. General parsing with regular expression matching. *Journal of Computer Languages*, 74:101176.

Juan Carlos Castillo, John E. Drury, and Kleanthes Grohmann. 2009. Merge over move and the extended projection principle. *Iberia*, 1(1).

Cristiano Chesi. 2021. Expectation-based minimalist grammars. *Computing Research Repository*, arXiv:2109.13871.

Noam Chomsky. 1961. On the notion 'rule of grammar'. In *Structure of Language and its Mathematical Aspects: Procs. 12th Symposium in Applied Mathematics*, pages 6–24.

Noam Chomsky. 1963. Formal properties of grammars. In R. Duncan Luce, Robert R. Bush, and Eugene Galanter, editors, *Handbook of Mathematical Psychology, Volume II*, pages 323–418. Wiley.

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press.

Noam Chomsky. 2000. Minimalist inquiries: The framework. In R. Martin, D. Michaels, and J. Uriagereka, editors, *Step by Step: Essays on Minimalism in Honor of Howard Lasnik*, pages 89–155. MIT Press.

Noam Chomsky. 2018. Syntactic structures. some retrospective comments. In Norbert Hornstein, Howard Lasnik, Pritty Patel-Grosz, and Charles Yang, editors, *Syntactic Structures after 60 Years*. De Gruyter Mouton.

Noam Chomsky. 2021. Minimalism: Where are we now, and where can we hope to go. *Gengo Kenkyu*, 160:1–41.

Noam Chomsky and George A. Miller. 1963. Introduction to the formal analysis of natural languages. In R. Duncan Luce, Robert R. Bush, and Eugene Galanter, editors, *Handbook of Mathematical Psychology, Volume II*, pages 269–321. Wiley.

Sandra Chung and James McCloskey. 1987. Government, barriers, and small clauses in Modern Irish. *Linguistic Inquiry*, 18(2):173–237.

Guglielmo Cinque. 2017. A microparametric approach to the head-initial/head-final parameter. *Linguistic Analysis*, 41.

Barbara Citko. 2011. Multidominance. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 119–142. Oxford University Press.

Chris Collins and Richard S. Kayne. 2020. Towards a theory of morphology as syntax. Technical report, New York University.

Robert Daland. 2014. What is computational phonology? *Loquens*, 1(1):e004.

Michael W. Daniels and W. Detmar Meurers. 2004. GIDLP: A grammar format for linearization-based HPSG. In *Procs. 11th Int. Conference on Head-Driven Phrase Structure Grammar*, page 93–111. CSLI.

Hossep Dolatian, Aniello De Santo, and Thomas Graf. 2021. Recursive prosody is not finite-state. *Procs 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, page 11–22.

Nigel Duffield. 1995. *Particles and Projections in Irish Syntax*. Springer.

Jason Eisner. 1997. Efficient generation in primitive optimality theory. In *Procs. 35th Annual Meeting of the Association for Computational Linguistics*.

Emily Elfner. 2012. *Syntax-Prosody Interactions in Irish*. Ph.D. thesis, University of Massachusetts, Amherst.

Mark T. Ellison. 1994. Phonological derivation in optimality theory. In *Procs. 15th Int. Conf. on Computational Linguistics*, pages 1007–1013.

Joost Engelfriet, Eric Lilin, and Andreas Maletti. 2009. Extended multi bottom–up tree transducers: Composition and decomposition. *Acta Informatica*, 46(8):561–590.

Samuel D. Epstein, Hisatsugu Kitahara, and T. Daniel Seely. 2012. Labeling by minimal search. *Linguistic Inquiry*, 45(13):463–481.

Marina Ermolaeva and Gregory M. Kobele. 2021. Agreement as information transmission over dependencies. Technical report, Universität Leipzig. Forthcoming.

Meaghan Fowlie. 2014. Adjunction and minimalist grammars. In *Formal Grammar: 19th International Conference*, pages 34–51. Springer.

Robert Frank and Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics*, 24:307–315.

Hans-Martin Gärtner. 2002. *Generalized Transformations and Beyond*. Akademie Verlag.

Hans-Martin Gärtner. 2014. Strange loops: Phrase-linking grammar meets Kaynean pronominalization. *Linguistische Berichte*, 2014(239):383–395.

Thomas Genet and Valérie Viet Triem Tong. 2001. Reachability analysis of term rewriting systems with Timbuk. In Robert Nieuenhuis and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence and Reasoning*. Springer LNAI 2250. Code: http://www.irisa.fr/lande/genet/timbuk/.

Jean-Yves Girard. 1987. Linear logic. *Theoretical Computer Science*, 50:1–102.

Kyle Gorman and Richard Sproat. 2021. *Finite-State Text Processing*. Morgan & Claypool.

Thomas Graf. 2012a. Concealed reference-set computation: How syntax escapes the parser's clutches. In Anna Maria Di Sciullo, editor, *Towards a Biolinguistic Understanding of Grammar. Essays on Interfaces*, pages 339–362. John Benjamins.

Thomas Graf. 2012b. Reference-set constraints as linear tree transductions via controlled optimality systems. In *Formal Grammar 2010/2011*, pages 97–113. Springer LNCS 7395. Slides.

Thomas Graf. 2018. Why movement comes for free once you have adjunction. In *Procs. Chicago Linguistic Society, CLS 53*, pages 117–136.

Thomas Graf. 2022. Subregular linguistics: Bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3-4):145–184.

Thomas Graf and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in minimalist grammars. In *Procs. Society for Computation in Linguistics*, volume 4, page 26.

Boris Harizanov and Vera Gribanova. 2019. Whither head movement? *Natural Language and Linguistic Theory*, 37:461–522.

Henk Harkema. 2001a. A characterization of minimalist languages. In *Logical Aspects of Computational Linguistics*, LNAI 2099, pages 193–211. Springer.

Henk Harkema. 2001b. *Parsing Minimalist Languages*. Ph.D. thesis, University of California.

Jeffrey Heinz and William J. Idsardi. 2017. Computational phonology today. *Phonology*, 34(2):211–219.

Jeffrey Heinz, Gregory M. Kobele, and Jason Riggle. 2009. Evaluating the complexity of optimality theory. *Linguistic Inquiry*, 40(2):277–288.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Procs 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Mans Hulden. 2009. FOMA: A finite-state compiler and library. In *Procs EACL 2009*.

Tim Hunter. 2011. Insertion minimalist grammars: Eliminating redundancies between merge and move. In *Procs Mathematics of Language 12, LNAI 6878*. Springer.

Tim Hunter. 2015. Deconstructing merge and move to make room for adjunction. *Syntax*, 18(3):266–319.

William J. Idsardi. 2018. Why is phonology different? No recursion. In Ángel J. Gallego and Roger Martin, editors, *Language, Syntax, and the Natural Sciences*, pages 212–223. Cambridge University Press.

Junko Ito and Armin Mester. 2012. Recursive prosodic phrasing in Japanese. In *Prosody Matters: Essays in Honor of Elisabeth Selkirk*, pages 280–303. Elsevier.

Gerhard Jäger. 2002. Gradient constraints in finite state OT: The unidirectional and the bidirectional case. In I. Kaufmann and B. Stiebels, editors, *More than Words: A Festschrift for Dieter Wunderlich*, pages 299–325. Akademie Verlag.

Trevor Jim and Yitzhak Mandelbaum. 2010. Efficient Earley parsing with regular right-hand sides. *Electronic Notes in Theoretical Computer Science*, 253:135–148.

Kyle Johnson. 2017. Rethinking linearization. Technical report, University of Massachussetts, Amherst.

Nicholas Kalivoda. 2018. *Syntax-Prosody Mismatches in Optimality Theory*. Ph.D. thesis, University of California, Santa Cruz.

Nicholas Kalivoda and Jennifer Bellik. 2020. Overtly headed XPs and Irish syntax–prosody mapping. In *Procs. Annual Meetings on Phonology*.

Laura Kallmeyer. 2010. *Parsing Beyond Context-Free Grammars*. Springer.

Makoto Kanazawa, Jens Michaelis, Sylvain Salvati, and Ryo Yoshinaka. 2011. Well-nestedness properly subsumes strict derivational minimalism. In *Logical Aspects of Computational Linguistics*, pages 112–128. Springer LNCS 6736.

Max Kanovich. 2015. Horn linear logic and Minsky machines. *Computing Research Repository*, arXiv:1512.04964.

Lauri Karttunen. 1998. The proper treatment of optimality in computational phonology. In *Procs International Workshop on Finite-State Methods in Natural Language Processing*.

Richard S. Kayne. 1994. *The Antisymmetry of Syntax*. MIT Press.

Richard S. Kayne. 2020. Antisymmetry and externalization. *LingBuzz*, 005554.

S. C. Kleene. 1956. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–42. Princeton University Press.

Gregory M. Kobele. 2019. Parsing ellipsis efficiently. In Robert C. Berwick and Edward P. Stabler, editors, *Minimalist Parsing*, pages 110–124. Oxford University Press.

Gregory M. Kobele. 2021. Minimalist grammars and decomposition. Technical report, Universität Leipzig. Forthcoming.

Yusuke Kubota and Robert D. Levine. 2021. *Type-Logical Syntax*. MIT Press.

Leland Kusmer. 2020. *Optimal Linearization: Prosodic Displacement in Khoekhoegowab and Beyond*. Ph.D. thesis, University of Massachusetts, Amherst.

Andrew Lamont. 2022. *Directional Harmonic Serialism*. Ph.D. thesis, University of Massachusetts, Amherst.

Howard Lasnik. 2011. What kind of computing device is the human language faculty? In A. M. di Sciullo and C. Boeckx, editors, *The Biolinguistic Enterprise*. Oxford University Press.

Howard Lasnik and Juan Uriagereka. 2022. *Structure: Concepts, Consequences, Interactions*. MIT Press.

Wim Martens and Joachim Niehren. 2005. Minimizing tree automata for unranked trees. In *Procs 10th Annual Symposium on Database Programming Languages*, pages 232–246. Springer LNCS 3774.

Jonathan May and Kevin Knight. 2006. Tiburon: A weighted tree automata toolkit. In *Proc. 11th International Conference on Implementation and Application of Automata*. Springer LNCS 4904. Code: https://github.com/isi-nlp/tiburon.

James McCloskey. 1999. On the right edge in Irish. *Syntax*, 2:189–209.

James McCloskey. 2002. Resumption, successive cyclicity, and the locality of operations. In Samuel D. Epstein and T. Daniel Seely, editors, *Derivation and explanation in the Minimalist Program*, pages 184–226. Blackwell.

James McCloskey. 2017. New thoughts on old questions – Resumption in Irish. In Jason Ostrove, Ruth Kramer, and Joseph Sabbagh, editors, *Asking the Right Questions: Essays in Honor of Sandra Chung*. University of California, Santa Cruz.

James McCloskey. 2022. The syntax of Irish Gaelic. Technical report, University of Califoria, Santa Cruz. Forthcoming.

Andrew McInnerney. 2022a. Against the argument/adjunct distinction. *Procs. 45th Annual Penn Linguistics Conference*, 28(1).

Andrew McInnerney. 2022b. *The Argument/Adjunct Distinction and the Structure of Prepositional Phrases*. Ph.D. thesis, University of Michigan.

Jens Michaelis. 2001. Transforming linear context free rewriting systems into minimalist grammars. In *Logical Aspects of Computational Linguistics*, pages 228–244. Springer LNCS 2099.

Daniel Milway. 2022. A parallel derivation theory of adjuncts. *Biolinguistics*, 16:e9313.

Richard Moot. 2002. *Proof Nets for Linguistic Analysis*. Ph.D. thesis, Utrecht University.

Richard Moot and Christian Retoré. 2012. *The Logic of Categorial Grammars*. Springer.

Glyn Morrill and Oriol Valentín. 2017. A reply to Kubota and Levine on gapping. *Natural Language and Linguistic Theory*, 35(1):257–270.

Ann E. Mulkern. 2003. *Cognitive Status, Discourse Salience, and Information Structure: Evidence from Irish And Oromo*. Ph.D. thesis, University of Minnesota.

Ann E. Mulkern. 2009. Left right behind: Irish pronoun postposing and information structure. In Andrew Carnie, editor, *Formal Approaches to Celtic Linguistics*. Cambridge Scholars.

Sara Myrberg. 2013. Sisterhood in prosodic branching. *Phonology*, 30(1):73–124.

Ad Neeleman, Joy Philip, Misako Tanaka, and Hans van de Koot. 2023. Subordination and binary branching. *Syntax*, Volume26(1).

Jairo Nunes. 1999. Linearization of chains and phonetic realization of chain links. In Samuel Epstein and Norbert Hornstein, editors, *Working Minimalism*, pages 217–249. MIT Press.

Kenji Oda. 2012. *Issues in the Left Periphery of Modern Irish*. Ph.D. thesis, University of Toronto.

Richard Pattis. 1994. Teaching EBNF first in CS 1. *ACM SIGCSE Bulletin*, 26(1):300–303.

Janet Pierrehumbert and Mary Beckman. 1988. *Japanese Tone Structure*. MIT Press.

Xavier Rival and Jean Goubault-Larrecq. 2001. Experiments with finite tree automata in Coq. In *Theorem Proving in Higher Order Logics*. Springer LNCS 2152. Code: https://github.com/coq-contribs/tree-automata.

Ian G. Roberts. 2021. *Parameter Hierarchies and Universal Grammar*. Oxford University Press.

Sylvain Salvati. 2011. Minimalist grammars in the light of logic. In S. Pogodalla, M. Quatrini, and C. Retoré, editors, *Logic and Grammar*. Springer LNCS 6700.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science*, 88:191–229.

Elisabeth Selkirk. 1996. The prosodic structure of function words. In J. Morgan and K. Demuth, editors, *Signal to Syntax*, pages 187–213. Lawrence Erlbaum.

Lisa Selkirk. 2011. The syntax-phonology interface. In *The Handbook of Phonological Theory*, pages 435–484. Wiley-Blackwell.

Stuart M. Shieber. 1984. Direct parsing of ID/LP grammars. *Linguistics and Philosophy*, 7(2):135–154.

Esturo Shima. 2000. A preference for move over merge. *Linguistic Inquiry*, 32(2).

Dominique Sportiche. 2017. Reconstruction, binding, and scope. In M. Everaert and H. van Riemsdijk, editors, *Blackwell Companion to Syntax, 2nd Ed.*

Edward P. Stabler. 1999. Remnant movement and complexity. In G. Bouma, E. Hinrichs, G. Kruijff, and D. Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Semantics*. CSLI.

Edward P. Stabler. 2001. Recognizing head movement. In P. de Groote, G. Morrill, and C. Retoré, editors, *Logical Aspects of Computational Linguistics*, LNAI 2099, pages 254–260. Springer.

Edward P. Stabler. 2011. Computational perspectives on minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–641. Oxford University Press.

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

Miloš Stanojević. 2019. On the computational complexity of head movement and affix hopping. In *Formal Grammar, 24th International Conference*, pages 101–116. Springer LNCS 11668.

Miloš Stanojević and Mark Steedman. 2021. Formal basis of a language universal. *Computational Linguistics*, 47(1):9–42.

John Torr and Edward P. Stabler. 2016. Coordination in minimalist grammars. In *Procs. 12th Workshop on Tree-Adjoining Grammars and Related Formalisms*.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax. *Procs. Society for Computation in Linguistics*, pages 267–276.

Philip Wadler. 1992. Comprehending monads. *Mathematical Structures in Computer Science*, 2(4):461–493.

Michael Wagner. 2010. Prosody and recursion in coordinate structures and beyond. *Natural Language and Linguistic Theory*, 28(1):183–237.

Joseph W. Windsor, Stephanie Coward, and Darin Flynn. 2018. Disentangling stress and pitch accent in Munster Irish. In *Procs 35th West Coast Conference on Formal Linguistics*. Cascadilla.

Niklaus Wirth. 1977. What can we do about the unnecessary diversity of notation for syntactic definitions? *Communications of the ACM*, 20(11):822–823.

Kristine M. Yu. 2021. Computational perspectives on phonological constituency and recursion. *Catalan Journal of Linguistics*, 77:114.

Kristine M. Yu. 2022. Representations for multiple dependencies in prosodic structures. *Proceedings of the Society for Computation in Linguistics*, 5:15.

A Weak equivalence of $*$-extensions: Sketch

Since $*$-MG extends MG, it is trivially true that $L(* - MG) \supseteq L(MG)$, and similarly for $*$-L(MCFG). Since $L(MG) = L(MCFG)$ (Harkema, 2001a; Michaelis, 2001), $L(*-MG) \subseteq L(MCFG)$ can be established by showing $L(* - MG) \subseteq L(MCFG)$. When labeling allows unbounded branching in the MG, a corresponding $*$-MCFG rule can be formulated. To construct a

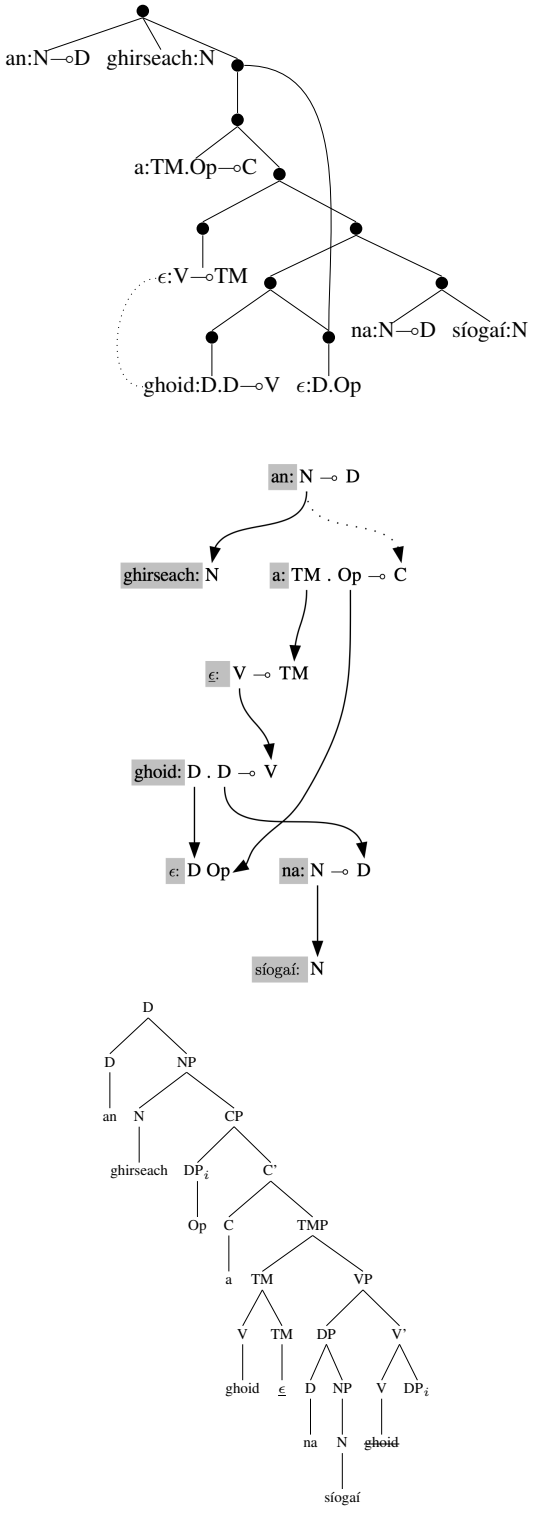


Figure 6: A set (leaves are lexical items, internal nodes are sets, arcs are $\in$ relations, with a dotted arc for head movement), dependency tree (with solid feature-checking arcs and dotted adjunct arc), and X-bar tree (for linguists) for (5a).

weakly equivalent MCFG, we simply replace unbounded branching with corresponding right recursive rules and prove the language is unchanged.

B Adjuncts and wh-movement

The approach used for coordination in the text is easily adapted to McInnerney (2022b)'s proposal, mentioned in the introduction, that unboundedly many adjuncts can be merged as sisters of the head they modify. His analysis is motivated in large part by a labeling theory that aims to reduce stipulated features, but as a place-holder for that kind of revision, here we simply extend our feature-based labeling to adjuncts.[15] It suffices to extend the definition of $\&(\gamma, \vec{C})$ in Figure 2 with one that is true whenever each element of C has a label of an admissible adjunct of γ.

In some dialects of Irish, when there is an Ā-extraction, as in the relative clause of (5a) from McCloskey (2002, (9)), the complementizer is pronounced differently than when there is resumption instead of extraction, as in (5b):[16]

(5) a. an ghirseach a ghoid na síogaí
the girl aL stole the fairies
'the girl that the fairies stole away'

b. an ghirseach a-r ghoid na síogaí í
the girl aN-[PAST] stole the fairies her
'the girl that the fairies stole away'

As a step towards MG implementation, let the relevant EPP/operator feature of aN be Op, in a relative clause adjoined as sister to the head N, in the structure for (5a) of Figure 6. Any number of additional adjuncts could occur as sister to the noun and relative clause.

C Implementation

Implementations of nondeterminism can be easy in programming languages like SWI Prolog that provide backtracking search. Represent {A,B} with the term `[A,B]` and $phon : a_1 \ldots a_i \multimap a_{i+1} \ldots a_{i+j}$ with `[phon]-[a1,...,ai]-[ai+1,...,ai+j]`. Then this 10 clause prolog implementation of R

[15] See McInnerney (2022b,a) on binding phenomena and other considerations that motivated the more common hierarchical analyses of adjunction. Cf. also Milway (2022); Graf (2018); Hunter (2015, 2011); Fowlie (2014).

[16] See McCloskey (2002, 2017); Oda (2012) and references cited there for careful discussion. Agreement and other relevant considerations are beyond the scope of this brief paper; see e.g. Ermolaeva and Kobele (2021) on agreement in an MG-based framework, Vu et al. (2019) on case.

calls itself recursively until complete structure A is generated from numeration X0, if possible:[17]

```
r([A], A) :- l(A, []-[_]-[]).
r(X0, X) :- select(A, X0, X1), l(A, [F0|AN]-AP-AC0),
    ( nonvar(F0), F0=p(F) -> P=true ; F0=F, P=fail),
    ( select([F|BP]-B, AC0, AC) -> X2 = X1, BC = []
    ; select(B, X1, X2), l(B, []-[F|BP]-BC), AC=AC0
    ), m(B, [F|AN]-AP-AC, []-[F|BP]-BC, ABF),
    '&'(P, F, X2, X3, Cs), mrg(A, B, Cs, ABCs),
    r([ABCs|X3], X).
mrg(A, B, Cs, [A,B|Cs]).
l(_-A-B, A-B-[]).
l([A,B], D) :- l(A, AF), l(B, BF), m(B, AF, BF, D).
m(B, [F0|AN]-AP-AC, BF, AN-AP-ABC) :-
 (nonvar(F0), F0=p(F) -> true ; F0=F),
 (select([F|BP]-B0, AC, AC1) -> B0=B,
   (BP = [] -> AC1 = ABC ;
    BP = [G|BP1], smc([[G|BP1]-B0|AC1], [], ABC, []))
 ; BF = []-[F]-BC, smc(AC, BC, ABC, [])
 ; BF = []-[F,G|BP]-BC, smc([[G|BP]-B|AC], BC, ABC, [])).
'&'(_,_,X,X,[]).
'&'(true,F,X0,X,[C|Cs]) :-
 select(C,X0,X1), l(C,[]-[F]-[]), '&'(true,F,X1,X,Cs).
smc([], D, D, _).
smc([[F|C]-A|L], M, [[F|C]-A|N], Fs) :-
    \+member(F, Fs), smc(L, M, N, [F|Fs]).
```

Derived structures here are lists not sets, but order of elements is irrelevant except for for identification of the head, and that is always determined by features alone. All syntactic structures in the text can be computed by this implementation. For example, this session computes the structure shown in Figure 3:

```
?- r([[]-[tm]-[c],[]-[v]-[tm],[is]-[a]-[v],
     [cuma]-[lpred]-[a],[]-[pred,d]-[lpred],
     [na]-[d]-[pred],[shamhradh]-[]-[d],
     [na]-[d]-[pred],[fhomar]-[]-[d],
     [na]-[d]-[pred],[gheimhreadh]-[]-[d],
     [no]-[X,p(X)]-[X],[e]-[]-[d]],A).

A = [
     []-[tm]-[c],
     [
       []-[v]-[tm],
       [
         [is]-[a]-[v],
         [
           [cuma]-[lpred]-[a],
           [
             [
               []-[pred,d]-[lpred],
               [
                 [
                   [no]-[pred,p(pred)]-[pred],
                   [
                     [na]-[d]-[pred],
                     [gheimhreadh]-[]-[d] ] ],
               [
                 [na]-[d]-[pred],
                 [fhomar]-[]-[d] ],
               [
                 [na]-[d]-[pred],
                 [shamhradh]-[]-[d] ] ] ],
         [e]-[]-[d] ] ] ] ] ].
```

As discussed in §4, efficiently implementing rule K, for parsing, requires more bookkeeping. In the deductive format of Stabler (2011, §A), for rule K, the feature checking rules for external merge (EM) with f^+ where $\sqcup$ is smc-respecting union:

$$\frac{s::f^+\alpha{-\!\circ}\beta,\gamma_1 \quad t\cdot f,\gamma_2}{st:\alpha{-\!\circ}\beta,\gamma_1\sqcup\gamma_2}\ (\text{EM1}^+)$$

$$\frac{s::f^+\alpha{-\!\circ}\beta,\gamma_1 \quad t\cdot f,\gamma_2}{st:f^+\alpha{-\!\circ}\beta,\gamma_1\sqcup\gamma_2}\ (\text{EM1}^{++}) \text{ if } t\neq\epsilon$$

$$\frac{s:f^+\alpha{-\!\circ}\beta,\gamma_1 \quad t\cdot f,\gamma_2}{ts:\alpha{-\!\circ}\beta,\gamma_1\sqcup\gamma_2}\ (\text{EM2}^+)$$

$$\frac{s:f^+\alpha{-\!\circ}\beta,\gamma_1 \quad t\cdot f,\gamma_2}{ts:f^+\alpha{-\!\circ}\beta,\gamma_1\sqcup\gamma_2}\ (\text{EM2}^{++}) \text{ if } t\neq\epsilon$$

$$\frac{s\cdot f^+\alpha{-\!\circ}\beta,\gamma_1 \quad t\cdot f\delta,\gamma_2}{s:\alpha{-\!\circ}\beta,\gamma_1\sqcup\gamma_2\sqcup\{t:\delta\}}\ (\text{EM3}^+) \text{ if } \delta\neq\epsilon.$$

Note that the Kleene + introduces indeterminacy, reflected here by the two rules for each of EM1 and EM2. The second case for external merge of a 'mover', EM3, and movement rules for these cases require a treatment of ATB movement – left for future work. The rules for f^* are the same, except that f^* is also 'checked' by 0 positive occurrences. See link in fn 17 for a working implementation.

The extension to rules for head movement can follow Stabler (2001); Stanojević (2019). The examples in the paper and the rules shown here only consider negative occurrences of f^+ and f^*. Positive occurrences may subsume previous proposals about 'persistent features', relevant for successive cyclic movement – left for future work.

The first steps toward a GEN transduction for prosody, discussed in §3, are also easily implemented. Represent a tree with root A and daughters B,C,D by the prolog term `A/[B,C,D]`. Then a relation that pairs the X-bar structure in Figure 4 – without coindexing and without the second coordinate – with the prosodic structure in Figure 1b, is computed by the following implementation:

```
head(X) :- member(X, [c,tm,v,a,lpred,pred,b]).
phrase(XP) :- atom_chars(XP, L), last(L, p).
gen(T, Out) :- rule(T, Out).
gen(X/L,T) :-  maplist(gen,L,S), rule(X/S,T).
rule(_/[qw/[X0], qphi/[X1]], qi/[i/[X0, X1]]).
rule(X/[Ph/[]], qw/[w/[Ph/[]]]) :- head(X).
rule(X/[Ph/[]], qphi/[phi/[w/[Ph/[]]]]) :- phrase(X).
rule(_/[], qe).
rule(_/[qw/[X0], qphi/[X1]], qphi/[phi/[X0,X1]]).
rule(_/[qphi/[X0], qphi/[X1]], qphi/[phi/[X0,X1]]).
rule(_/[qe, X0], X0).
rule(_/[X0, qe], X0).
rule(_/[X0], X0).
```

Representing the reduced Figure 4 by a prolog term, as the first argument to `gen`, this code computes the prolog term for Figure 1b as the first of many candidate structures.

[17] This code (with some explanatory comments!) is available at https://github.com/epstabler/star, along with display tools, a parser for rule K (in python), and tree transducers.