

Preface: SCiL 2024 Editors' Note

Richard Futrell, Connor Mayer & Noga Zaslavsky
University of California, Irvine

This volume contains research presented at the sixth annual meeting of the Society for Computation in Linguistics (SCiL), held in Irvine, California, June 27–29, 2024.

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 69 submissions to the conference. 20 submissions were selected for oral presentation (~29%) and 29 for poster presentation (~42%).

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

Eric Raimy, Shane Steinert-Threlkeld, Dongsung Kim, Andrew Lamont, Bonnie L. Webber, Charlie OHara, Aniello De Santo, Laurel Perkins, Edward P. Stabler, Robert Malouf, Caitlin Smith, Richard Futrell, Eric Rosen, Jonathan Brennan, Xiaoli Chen, Joseph Pater, Brian Dillon, Thomas Graf, Kyle Gorman, Ollie Sayeed, Gaja Jarosz, Lindy Comstock, Rui Chaves, Emily Morgan, Timothee Mickus, R. Thomas McCoy, Yang Janet Liu, Canaan Breiss, Jordan Kodner, Maayan Keshev, Zoey Liu, Jonathan Rawski, Qihui Xu, Dylan Bumford, Kyle Mahowald, Roger Levy, Tamar Johnson, Yohei Oseki, Robert Frank, Tiago Pimentel, Adam Jardine, Vsevolod Kapatsinski, Caleb Belth, Shiva Upadhye, Hossep Dolatian, Giorgio Magri, Shuge Lei, Laura Gwilliams, Frederic Mailhot, Brandon Prickett, Sebastian Schuster, Christopher Potts, Sheng-Fu Wang, Philippe de Groote, Itamar Kastner, Cassandra Jacobs, Tim Hunter, Christo Kirov, Shohini Bhattachali, Carolyn Jane Anderson, Alexander Clark, Tracy Holloway King, Cory Shain, Katrin Erk, Francis Mollica, Connor Mayer, Colin Wilson, Lisa Pearl, Clara Isabel Meister, Olga Zamaraeva, Ollie Sayeed, Eric Rosen, Joe Pater, and Agata Wolna.

Thanks also to Joe Pater, Brandon Prickett, Tim Hunter, Erin Jerome, Yanting Li, Shiva Upadhye, Jiaxuan Li, Nathaniel Imel, Niels Dickson, Noa Attali and Yuting Gu for logistical help.

We gratefully acknowledge the sponsorship of the UCI School of Social Sciences, UCI Center for Theoretical Behavioral Sciences, UCI Donald Bren School of Information & Computer Sciences, and Google, Inc.

SCiL 2024 also included invited talks by Noah Goodman (Stanford University), Jenny Culbertson (University of Edinburgh), and Jacob Andreas (Massachusetts Institute of Technology). Further information can be found at our website: <https://sites.uci.edu/scil2024/>.

Tensor Product Representations of Regular Transductions

Zhouyi Sun¹ Jonathan Rawski^{1,2}

¹Massachusetts Institute of Technology ²San Jose State University
{szy, rawski}@mit.edu

Abstract

This paper provides a vector space characterization of regular transductions. We use finite model theory to characterize objects like strings and trees as relational structures and origin graphs to characterize input-output relations generated by transducer. We show detailed processes of using multilinear maps as function application for evaluation to compile regular transductions characterized by MSO definable origin graphs into a tensor embedding.

1 Introduction

The mathematical theory of automata provides a way to explicitly tie the complexity of linguistic patterns to specific claims about memory organization and thus provides an direct way of measuring the cognitive demands of language. Transducers, i.e. automata that produce outputs beyond “yes” or “no”, have been around since the beginnings of automata theory, and have a long history in linguistics and NLP for modeling the complexity of various linguistic processes (Mohri, 1997; Heinz, 2018; Roark and Sproat, 2007).

One particular class of interest is the regular transductions, which generalize the class of regular languages. Regular languages are one of the most well-studied objects in computer science, characterized by regular expressions, finite-state automata, and statements in Monadic Second-Order logic, among others (Thomas, 1997). Linguistically, the regular class has been shown to sufficiently characterize phonological and morphological phenomena (Kaplan and Kay, 1994; Rawski et al., 2023; Doltan et al., 2021).

The regular transductions have become far better understood in recent years. Engelfriet and Hoogboom (2001) showed that MSO-transducers, a logical model of transducers studied in the general context of graph transductions (Courcelle, 1994; Courcelle and Engelfriet, 2012), exactly characterize the transductions realized by two-way transduc-

ers. A model of one-way transducers with registers, called streaming string transducers, has also been shown to capture the same class of transductions, which were then called regular functions (Alur and Černý, 2010).

This paper considers logical characterizations of regular functions over structures that are defined using finite model theory. Model theory has been used for comparisons of particular grammatical theories in phonology and syntax (Rogers, 1998; Pullum, 2007; Graf, 2010), and for studying the nature of linguistic structures and processes themselves (Heinz, 2018; Payne et al., 2016). Linguistic structures like strings and trees are modeled using relational information which holds among the elements characterizing a particular structure.

It is of interest to see how these models may be characterized in vector spaces. Vector space approaches to language and symbolic cognition in general have become increasingly popular during the last two decades. There is work dealing with conceptual spaces for sensory representations (Gardenfors, 2004), multilinear representations for compositional semantics (Blutner, 2009; Aerts, 2009), and dynamical systems for modeling language processes (Beim Graben et al., 2008; Tabor, 2009).

One particularly significant contribution in this area is Tensor Product Representation (Smolensky, 1990). Here, subsymbolic dynamics of neural activation patterns in a vector space description become interpreted as symbolic cognitive computations at a higher-level description by means of “filler/role” bindings via tensor products. These tensor product representations form the symbolic foundation of Harmonic Grammar and Optimality Theory, and have been successfully employed for phonological and syntactic computations (Smolensky and Legendre, 2006).

Tensor methods and (sub)regular grammars/automata have been used to evaluate and interpret neural networks (Rabanser et al., 2017).

McCoy et al. (2018) showed that recurrent neural networks (RNNs) implicitly encode tensor product representations, and Strobl et al. (2023) survey work using regular languages to test transformer language models. Nelson et al. (2020) used regular string transductions to test the generalization capacity of RNNs, finding that they failed to successfully learn them unless explicitly given machinery which enabled them to approximate the underlying two-way finite-state transducer.

There has also work on embedding logical calculi using tensors. Grefenstette (2013) introduces tensor-based predicate calculus that realizes logical operations. Yang et al. (2014) introduce a method of mining Horn clauses from relational facts represented in a vector space. Serafini and Garcez (2016) introduce logic tensor networks that integrate logical deductive reasoning and data-driven relational learning. Sato (2017) formalizes Tarskian semantics of first-order logic in vector spaces. Rawski (2019) employed Sato’s method to translate model-theoretic representations and languages definable in first-order logic (the star-free and locally threshold testable sets) into tensors. This paper extends that work to consider transductions.

2 Transductions as origin graphs

Model theory, combined with logic, provides a powerful way to study and understand mathematical objects with structures (Enderton, 2001). This paper only considers finite relational models (Libkin, 2004).

Definition 1. A model signature is a tuple $S = \langle D; R_1, R_2, \dots, R_m \rangle$ where the domain D is a finite set, and each R_i is a n_i -ary relation over the domain.

In this paper, the relations are at most binary.

Definition 2. A model for a set of objects M is a total, one-to-one function from M to structures whose type is given by a model signature S .

The flexibility given by model-theoretic representations allows them to consider many things as objects, including relations between inputs and outputs. Bojańczyk (2014), attempting to solve the problem of how to decide whether two transducers generate the same input-output pairs, created a novel way to describe transductions as model-theoretic structures by using an *origin mapping*. Informally, an origin mapping, which is a total function from output positions to input positions showing which input position(s) are used for a given

output symbol. Bojańczyk et al. (2017) directly considered this relation between inputs and outputs as a structure, called an *origin graph*, for which they defined the corresponding idea of an *origin transduction*.

Definition 3. (Bojańczyk et al., 2017) An origin string-to-string transduction (origin transduction for short) consists of an input alphabet Σ , an output alphabet Γ , and a set of origin graphs over these alphabets specifying the input position used to produce each output position.

An origin graph with input w and output v consists of:

- The domain which is the disjoint union of positions in w and positions in v ;
- Two binary predicates for the successor relations in w and v ;
- a binary predicate, the origin mapping, which is a total function from output position to input positions;
- a unary predicate for each $a \in \Sigma \cup \Gamma$ which identifies positions with label a .

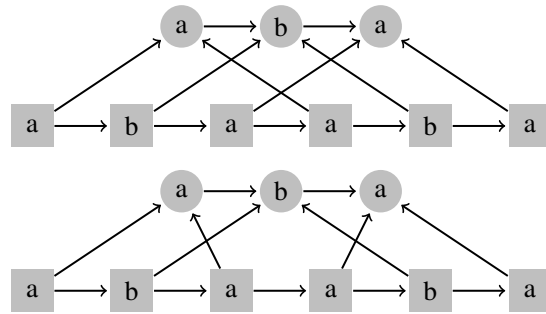


Figure 1: Visualizations of two string-to-string transductions differing only on the origin mapping.

Consider two transductions from the input string aba to the output string $abaaba$ visualized in fig. 1. In addition to the pair of input and output string ($aba \rightarrow abaaba$) involved in the transduction, the origin input position of each output position differentiates origin transductions, especially in terms of their recognizability by automata (see e.g. Dolatian et al., 2021). In fig. 1, input positions are denoted by circles. Output positions are denoted by squares. And the arrows from an output position to an input position represents the origin information of the output position. The upper figure can be seen

as the result of reduplication computed by a two-way automaton which goes back-and-forth over the input.

Origin graphs present an intriguing way to take concepts used to define classes of formal languages, and apply them to transductions by considering them as structures. This has a history in linguistics, namely through evaluations of phonological ideas. Various recent works have used formalisms resembling origin semantics to discuss Correspondence Theory (Payne et al., 2016) and rewrite-rule interaction (Meinhardt et al., 2024).

3 Logical Languages and Transduction Classes

Usually a model signature provides the vocabulary for some logical language \mathcal{L} , which contains N constants $\{e_1, \dots, e_N\}$. Following notation of Sato (2017), a model $M = (D, I)$ is thus a pair of domain, a nonempty set D and an interpretation I that maps constants e_i to elements (entities, individuals) $I(e_i) \in D$ and k -ary predicate symbols r to k -ary relations $I(r) \subseteq D^k$.

An assignment a is a mapping from variables x to an element $a(x) \in D$. It provides a way of evaluating formulas containing free variables. Syntactically terms mean variables and/or constants and atomic formulas or atoms $r(t_1, \dots, t_k)$ are comprised of a k -ary predicate symbol r and k terms t_1, \dots, t_k some of which may be variables. Formulas F in \mathcal{L} are inductively constructed as usual from atoms using logical connectives (negation \neg , conjunction \wedge , disjunction \vee) and quantifiers (\exists, \forall). First-order formulas allow only quantification over elements. Monadic Second-Order (MSO) formulas additionally allow countably many second-order set variables X, Y, \dots with $x \in X$, which can be quantified over $\forall X, \exists X$. In this case, a is a mapping from set variables variables X to a set of elements $a(X) \in D$.

Sentences in this logical language define sets of strings/trees as follows. The language of a formula F is all and only those graphs whose models satisfy F . For any formula F , $\llbracket F \rrbracket_{I,a} \in \{1, 0\}$ and when $\llbracket F \rrbracket_{I,a} = 1$, we write $M \models_a F$ to mean the model satisfies F . However when F is closed, since $\llbracket F \rrbracket_{I,a}$ does not depend on the assignment a , we just write $\llbracket F \rrbracket$ and $M \models F$ if F is true in M .

There are several well-known connections between logical statements and languages classes. Most famous is Büchi (1960)'s result that lan-

guages characterizable by finite-state machines, the regular languages, are equivalent to statements in Monadic Second-Order Logic over the precedence model for strings (and successor, since precedence is MSO-definable from successor).

Courcelle (1994) lifted the idea of MSO to transductions, creating the MSO-definable analog of the regular languages. Engelfriet and Hoogeboom (2001) showed an equivalence between MSO-transducers and two-way transducers (where the read head can move back and forth on the input). Later, Alur and Černý (2010) showed another equivalence with streaming string transducers (where the two-way read head is replaced with a finite number of registers), giving the following result:

Theorem 1 (Engelfriet and Hoogeboom, 2001; Alur and Černý, 2010; Courcelle, 1994). *A transduction is regular iff it is realized by a 2-way Deterministic Finite-state Transducer or an MSO-transducer or a Streaming String Transducer.*

This convergence of results led to particular problems of deciding whether a given transducer is equivalent to another one, or whether two transducers compute the same string relation in the same way. This is analogous to the concept of weak versus strong capacity in linguistics (Miller, 1999). Bojańczyk et al. (2017), using the origin graphs defined earlier, showed another characterization:

Theorem 2. (Bojańczyk et al., 2017) *Let G be an origin transduction, i.e., an input alphabet, an output alphabet, and a set of origin graphs over these alphabets. G is a regular function (recognized by a streaming string transducer) iff:*

bounded origin: *there is some $m \in \mathbb{N}$ such that in every origin graph from G , every input position is the origin of at most m output positions;*

k -crossing: *in every origin graph from \mathcal{G} , every input position is crossed by at most k output positions (An output position j crosses an input position i if the origin of j is no greater than i , and either j is the final output position, or the successor of j has its origin greater than i .);*

MSO-definable: *there is an MSO formula which is true in exactly the origin graphs from \mathcal{G} .*

The main goal of this paper, extending Rawski (2019), is to characterize the origin graphs with these properties via tensor calculus in order to embed transductions in vector spaces.

4 Tensor Representations of Logical Constraints

We first want to show how to embed a model domain and signature into a vector space, using tensors to encode relational information. We then show how the ingredients of a logical language (specifically, First-Order and Monadic Second-Order logics) translate to operations over tensors.

Scalars are denoted with lower case letters like a . Vectors mean column vectors and we denote them by boldface lower case letters like \mathbf{a} and \mathbf{a} 's components by a_i . $\mathcal{D}' = \{\mathbf{e}_1, \dots, \mathbf{e}_N\}$ is the standard basis of N -dimensional Euclidean space \mathbb{R}^N where $\mathbf{e}_i = (0 \dots, 1, \dots, 0)^T$ is a vector that has one at the i -th position and zeros elsewhere. Such vectors are called one-hot vectors. For set variables are denoted

$\mathbf{1}$ is a vector of all ones. We assume square matrices, written by boldface upper case letters like \mathbf{A} . \mathbf{I} is an identity matrix, and $\mathbb{1}$ is a matrix of all ones. Order- p tensors $\mathcal{A} \in \mathbb{R}^{D^p}$, are also denoted by $\{a_{i_1, \dots, i_p}\}$ ($1 \leq i_1, \dots, i_p \leq N$). \mathcal{A} 's component a_{i_1, \dots, i_p} is also written as $(\mathcal{A})_{i_1, \dots, i_p}$. $(\mathbf{a} \bullet \mathbf{b}) = \mathbf{a}^T \mathbf{b}$ is the inner product of \mathbf{a} and \mathbf{b} whereas $\mathbf{a} \circ \mathbf{b} = \mathbf{a} \mathbf{b}^T$ is their outer product. $\mathbf{1} \circ \dots \circ \mathbf{1}$ is a k -order tensor, and $\mathbf{1} \circ \dots \circ \mathbf{1} (\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k}) = (\mathbf{1} \bullet \mathbf{e}_{i_1}) \dots (\mathbf{1} \bullet \mathbf{e}_{i_k}) = 1$. Scalars, vectors, and matrices are tensors of order 0, 1, and 2 respectively.

There exists an isomorphism between tensors and multilinear maps (Bourbaki, 1989), such that any carried multilinear map

$$f : V_1 \rightarrow \dots \rightarrow V_j \rightarrow V_k$$

can be represented as a tensor $\mathcal{T}_f \in V_k \otimes V_j \otimes \dots \otimes V_1$. This means that tensor contraction acts as function application. This isomorphism guarantees that there exists such a tensor \mathcal{T}^f for every f , such that for any $v_1 \in V_1, \dots, v_j \in V_j$:

$$f \mathbf{v}_1 \dots \mathbf{v}_j = \mathbf{v}_k = \mathcal{T}^f \times \mathbf{v}_1 \times \dots \times \mathbf{v}_j \quad (1)$$

Following Sato (2017), we first isomorphically map a model M to a model M' in \mathbb{R}^N . We map entities $e_i \in D$ to one-hot vectors $\{\mathbf{e}_i\}$. So D is mapped to $D' = \{\mathbf{e}_1, \dots, \mathbf{e}_N\}$, the basis of \mathbb{R}^N . We next map a k -ary relation r in M to a k -ary relation r' over D' which is computed by an order- k tensor $\mathcal{R} = \{r_{i_1, \dots, i_k}\}$, whose truth value $\llbracket r(e_{i_1}, \dots, e_{i_k}) \rrbracket$ in M is given by

$$\llbracket r(e_{i_1}, \dots, e_{i_k}) \rrbracket$$

$$\begin{aligned} &= \mathcal{R}(\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k}) \\ &= \mathcal{R} \times_{\mathbf{1}} \mathbf{e}_{i_1} \times_{\mathbf{1}} \dots \times_{\mathbf{1}} \mathbf{e}_{i_k} \\ &= r_{i_1, \dots, i_k} \in \{1, 0\} \quad (\forall i_1, \dots, i_k \in \{1, \dots, N\}) \end{aligned} \quad (2)$$

We identify r' with \mathcal{R} so that \mathcal{R} encodes the M -relation r . Let M' be a model (D', I') in \mathbb{R}^N such that I' interprets entities by $I'(e_i) = \mathbf{e}_i$ ($1 \leq i \leq N$) and relations r by $I'(r) = \mathcal{R}$.

For the purposes of this paper, we restrict ourselves to binary relations and predicates. When r is a binary predicate, the corresponding tensor \mathcal{R} is a bilinear map and represented by an adjacency matrix \mathbf{R} as follows:

$$\llbracket (e_i, e_j) \rrbracket = (\mathbf{e}_i \cdot \mathbf{R} \mathbf{e}_j) = \mathbf{e}_i^T \mathbf{R} \mathbf{e}_j = r_{ij} \in \{1, 0\} \quad (3)$$

Note that when $r(x, y)$ is encoded by \mathcal{R} as $(\mathbf{x} \bullet \mathbf{R} \mathbf{y})$, $r(y, x)$ is encoded by \mathbf{R}^T , since $(\mathbf{y} \bullet \mathbf{R} \mathbf{x}) = (\mathbf{x} \bullet \mathbf{R}^T \mathbf{y})$ holds

We next inductively define the evaluation $\llbracket F \rrbracket_{I', a'}$ of a formula F in M' . Let a be an assignment in M and a' the corresponding assignment in M' , so $a(x) = e_i$ iff $a'(x) = \mathbf{e}_i$. For a ground atom $r(e_{i_1}, \dots, e_{i_k})$, define

$$\llbracket r(e_{i_1}, \dots, e_{i_k}) \rrbracket' = \underline{\mathbf{R}}(\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k}) \quad (\forall i_1, \dots, i_k \in \{1, \dots, N\}). \quad (4)$$

where $\mathcal{R} = \{r_{i_1, \dots, i_k}\}$ is a tensor encoding the M -relation r in M . By definition $\llbracket F \rrbracket_{I, a} = \llbracket F \rrbracket_{I, a}$ holds for any atom F . Negative literals are evaluated using $\neg \mathcal{R}$ defined as

$$\llbracket \neg r(e_{i_1}, \dots, e_{i_k}) \rrbracket' = \neg \mathcal{R}(\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k}) \quad (5)$$

$$\text{where } \neg \mathcal{R} \stackrel{\text{def}}{=} \overbrace{\mathbf{1} \circ \dots \circ \mathbf{1}}^k - \mathcal{R}$$

$\neg \mathcal{R}$ encodes an M -relation $\neg r_1$. Negation other than negative literals, conjunction, disjunction, and quantifiers are evaluated in M' as follows.

$$\llbracket \neg F \rrbracket' = 1 - \llbracket F \rrbracket' \quad (6)$$

$$\llbracket F_1 \wedge \dots \wedge F_h \rrbracket' = \llbracket F_1 \rrbracket' \dots \llbracket F_h \rrbracket' \quad (7)$$

$$\llbracket F_1 \vee \dots \vee F_h \rrbracket' = \min_1(\llbracket F_1 \rrbracket' + \dots + \llbracket F_h \rrbracket') \quad (8)$$

$$\llbracket \exists y F \rrbracket' = \min_1 \left(\sum_{i=1}^N \llbracket F_{y \leftarrow e_i} \rrbracket' \right) \quad (9)$$

Here the operation $\min_1(x) = \min(x, 1) = x$ if $x < 1$, otherwise 1, as componentwise application. $F_{y \leftarrow e_i}$ means replacing every free occurrence

of y in F with e_i . Universal quantification over individual elements is treated as $\forall x F = \neg \exists x \neg F$.

Monadic second order logic allows variables over sets of elements in addition to first order formulas discussed above. A set variable X consisting of k entities $e_{i_1}, e_{i_2}, \dots, e_{i_k}$ can be represented as the sum of the corresponding one-hot vectors, $e_X = \sum_{e_i \in X} e_i$ which is a k -hot vector. Notice here the subscript is a set of numbers X instead of a number indicating the position, e.g. i . The evaluation of ground atoms like $r(E_{i_1}, \dots, E_{i_k})$ can stay the same, which is also true of other first order evaluations. The existential quantification of set variables can be evaluated in M' as follows.

$$\llbracket \exists X F \rrbracket' = \min_1 \left(\sum_{I \subseteq D} \llbracket F_{X \leftarrow I} \rrbracket' \right) \quad (10)$$

Similarly, universal quantification over sets can be treated as $\forall X F = \neg \exists X \neg F$.

We can now define the properties over origin graphs using this formulation as in Theorem 2. For an origin graph, the set of its input positions is N and the set of its output positions is M . Binary relation R_{origin} defines the origin information between N and M . $R_{\text{origin}}(i, j) = 1$ when the output position j has the input position i as its origin.

For every input position x , the condition of k -crossing is equivalent to

$$\sum_{i=1}^{|M|} R_{\text{origin}}(x, i) \leq k \quad (11)$$

The condition of bounded origin is equivalent to

$$\sum_{i=1}^{|M|} \text{cross}(x, i) \leq k \quad (12)$$

Equating input positions with the natural numbers 1 to $|N|$ and output positions with natural numbers 1 to $|M|$, the function *cross* (see theorem 2 for definition) can be defined as

$$\text{cross}(x, i) = \begin{cases} 1 & \exists k (k \leq x \wedge R(k, i)) \wedge \\ & \wedge (i = |M| \vee \exists l (x < l \wedge \\ & \wedge R(l, i + 1))) \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, $\text{cross}(x, i)$ returns 1 if an output position i has origin at some input position k preceding an input position x , such that i is either the last output position or its successor has origin to the right of x . It returns 0 otherwise.

5 Examples

This section presents detailed processes of compiling MSO definable origin graphs into a tensor embedding, based on Sato (2017); Rawski (2019). MSO formulas are first converted into prenex normal form. Then the formula is translated into its corresponding tensor representation by applying evaluation rules discussed in section 4. For readability, we often collapse multiple sequential \exists quantifiers into one.

5.1 -t insertion

We begin with a simple process of concatenating a symbol $-t$ onto the end of an output word, akin to suffixation or epenthesis. This process of $-t$ insertion maps, for example, the input string ba to bat . The process can be captured by an MSO formula (in fact, a first order formula):

$$\begin{aligned} F_{-t} = & \forall x (R_{\text{input}}(x) \rightarrow \\ & \exists y (R_{\text{origin}}(x, y) \wedge R_{\text{equal}}(x, y) \wedge ((R_{\text{last-i}}(x) \wedge \\ & \exists z (R_{\text{succ-o}}(y, z) \wedge R_{\text{origin}}(x, z) \wedge R_{\text{t-o}}(z) \wedge R_{\text{last-o}}(z))) \\ & \vee \exists x', y' (R_{\text{succ-i}}(x, x') \wedge R_{\text{succ-o}}(y, y') \wedge \\ & \wedge R_{\text{origin}}(x', y')))) \quad (13) \end{aligned}$$

We assert that for any input position x ($R_{\text{input}}(x) = 1$), there exists an output position y whose origin is x ($R_{\text{origin}}(x, y) = 1$) and the labels of x and y are the same ($R_{\text{equal}}(x, y) = 1$). We follow Bojańczyk et al. (2017) in distinguishing the set of input alphabet and the set of output alphabet. For example, $R_{\text{t-output}}(z)$ checks whether position z is an output position and whether its label is t . In this way, unary predicates R_{input} , R_{output} and the binary predicate R_{equal} can all be defined with no difficulty.

Additionally, if x is the last input position ($R_{\text{last-i}}(x) = 1$), then y has a successor z . The origin of z is x ($R_{\text{origin}}(x, z) = 1$). Its label is t in the output alphabet ($R_{\text{t-output}}(z) = 1$). And it is the last position in the output ($R_{\text{last-o}}(z) = 1$). Otherwise x has its successor x' ($R_{\text{succ-i}}(x, x') = 1$) and y also has its successor y' ($R_{\text{succ-o}}(y, y') = 1$), whose origin is x' ($R_{\text{origin}}(x', y') = 1$).

The adjacency matrix defined by the binary relation R_{origin} in this case is almost an identity matrix. Suppose the input has a length of n . Then for any i less than n , (i, i) is 1. $(n, n + 1)$ is also 1. All other entries are 0. It satisfies the constraints of **bounded origin** and **k-crossing** naturally, as each

input position is the origin of at most 2 output positions and is crossed by at most 2 output positions. And thus the origin transduction of $-t$ insertion is recognizable by a streaming string transducer with k registers.

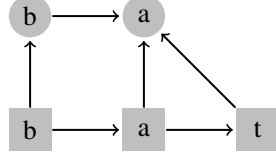


Figure 2: Visualizations of the origin graph of suffixiating $-t$ to ba .

Because the formula of $-t$ or any suffix insertion is properly first order (in fact it's subsequential (Mohri, 1997)), its origin graph has a straightforward embedding into tensor operations following results from Rawski (2019) (see also section 4).

First, the formula can be converted into prenex normal form:

$$\begin{aligned} & \forall x \exists y \exists z \exists x' \exists y' (\neg R_{\text{input}}(x) \vee (R_{\text{origin}}(x, y) \wedge \\ & \quad \wedge R_{\text{equal}}(x, y) \wedge ((R_{\text{last-i}}(x) \wedge \\ & \wedge (R_{\text{succ-o}}(y, z) \wedge R_{\text{origin}}(x, z) \wedge R_{\text{t-o}}(z) \wedge R_{\text{last-o}}(z))) \vee \\ & \vee (R_{\text{succ-i}}(x, x') \wedge R_{\text{succ-o}}(y, y') \wedge R_{\text{origin}}(x', y')))) \end{aligned} \quad (14)$$

Compiling the prenex formal formula into tensor notation, we get

$$\begin{aligned} T_{-t} = & 1 - \min_1 \sum_{x=1}^N (1 - \min_1 \sum_{y,z,x',y'=1}^N \\ & (\min_1 ((1 - \mathcal{R}^{\text{input}} e_x) + (e_x^T \mathcal{R}^{\text{origin}} e_y) \bullet (e_x^T \mathcal{R}^{\text{equal}} e_y) \\ & \quad \bullet \min_1 ((\mathcal{R}^{\text{last-i}} e_x) \bullet (e_y^T \mathcal{R}^{\text{succ-o}} e_z) \\ & \quad \bullet (e_x^T \mathcal{R}^{\text{origin}} e_z) \bullet (\mathcal{R}^{\text{t-i}} e_z) \bullet (\mathcal{R}^{\text{last-o}} e_z) + \\ & ((e_x^T \mathcal{R}^{\text{succ-i}} e_{x'}) \bullet (e_y^T \mathcal{R}^{\text{succ-o}} e_{y'}) \bullet (e_{x'}^T \mathcal{R}^{\text{origin}} e_{y'} \\ &)))))) \end{aligned} \quad (15)$$

Here we can see how each of the ingredients of the logical formula maps, straightforwardly, to ingredients of the tensor formulation. Note that $\sum_{y,z,x',y'=1}^N$ collapses the four existential quantifiers, for ease of readability.

5.2 Copying

Next we demonstrate an MSO formula for the process of copying the input word, which is of more

linguistic significance. Copying, known as reduplication in linguistics, is a common morphological process which is often argued to be among the most complex phenomena in linguistics. Copying is properly a regular function, and is one of the standard characteristic functions used to define the properties of the class, namely the linear growth property (see Rawski et al. (2023) for details).

$$\begin{aligned} F_{\text{copying}} = & \forall x (R_{\text{input}}(x) \rightarrow \\ & \rightarrow \exists Y, Z (R_{\text{output-path}}(Y, Z) \wedge \\ & \wedge \exists y, z (R_{\text{origin}}(x, y) \wedge R_{\text{origin}}(x, z) \wedge y \in Y \wedge z \in Z \wedge \\ & \quad \wedge (\neg R_{\text{last-input}}(x) \rightarrow \\ & \quad \exists x', y', z' (R_{\text{succ-input}}(x, x') \wedge R_{\text{succ-output}}(y, y') \wedge \\ & \quad \wedge R_{\text{succ-output}}(z, z') \wedge R_{\text{origin}}(x', y') \wedge R_{\text{origin}}(x', z') \\ & \quad)))) \end{aligned} \quad (16)$$

The binary predicate $R_{\text{output-path}}(Y, Z)$ holds when Y and Z partition the output positions, in which Y and Z are two paths and the first position of Z is the successor of the last position of Y . It can be formalized by the conjunction of three predicates: 1) Y and Z together cover all output positions, with each position belonging exclusively to either Y or Z ; 2) Both Y and Z must be paths. A path is defined as a connected graph where every position, except one, has a successor within the path, and every position, except one, is a successor of another position in the path; 3) the tail of Y has the head of X as its successor in X (the tail of a path can be defined as the only position without a successor in the path; the head of a path is not a successor of any position in the path).

The formula thus states that for every input position x , there are two output positions $y \in Y$ and $z \in Z$, whose origin is x . The part requires they bear the same label is omitted for the ease of understanding. Whenever x is not the last position in the input, y and z are also not last positions in Y and Z . Their successors y' and z' both have the successor of x , x' as their origin.

In this way, each input position is mapped to exactly two output positions and **bounded origin** is satisfied. The last input position is only crossed once by the last output position. Every other input position is crossed twice by the two output positions it mapped to. And therefore **k-crossing** is satisfied as well. Importantly, the number of copies must be linearly bounded (i.e. some pre-specified number of copies of an input string) in order to remain MSO-definable. Unbounded copying vio-

lates both constraints and is not MSO definable, nor is bounding the number of copies to some higher order, say polynomially. (Rawski et al., 2023).

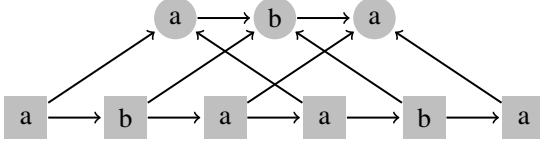


Figure 3: Visualizations of the origin graph of copying *aba*

Converting the formula into prenex normal form:

$$\begin{aligned}
F_{\text{copying}} = & \forall x \exists Y \exists Z \exists y \exists z \exists x' \exists y' \exists z' (\neg R_{\text{INPUT}}(x) \\
& \vee (R_{\text{output-path}}(Y, Z) \\
& \wedge (R_{\text{origin}}(x, y) \wedge R_{\text{origin}}(x, z) \wedge y \in Y \wedge z \in Z \\
& \wedge (R_{\text{last-input}}(x) \vee (R_{\text{succ-input}}(x, x') \wedge R_{\text{succ-output}}(y, y') \\
& \wedge R_{\text{succ-output}}(z, z') \wedge R_{\text{origin}}(x', y') \wedge R_{\text{origin}}(x', z') \\
&)))) \quad (17)
\end{aligned}$$

The formula can be compiled into tensor notation as follows:

$$\begin{aligned}
\mathcal{T}_{\text{copying}} = & \\
& 1 - \min_1 \left(\sum_{x=1}^N (1 - \min_1 \sum_{Y, Z \subseteq D} (\min_1 \sum_{y, z, x', y', z'=1}^N \right. \\
& \min_1 ((1 - \mathcal{R}^{\text{origin}} e_{y'}) + (\mathcal{R}^{\text{succ-output}} \times e_Y \times e_Z) \bullet \\
& (e_x^T \mathcal{R}^{\text{origin}} e_y) \bullet (e_x^T \mathcal{R}^{\text{origin}} e_z) \bullet (e_y \bullet e_Y) \bullet (e_y \bullet e_Y) \bullet \\
& \min_1 (\mathcal{R}^{\text{last-input}} e_x + (e_x^T \mathcal{R}^{\text{succ-input}} e_{x'}) \bullet \\
& (e_y^T \mathcal{R}^{\text{succ-output}} e_{y'}) \bullet (e_z^T \mathcal{R}^{\text{succ-output}} e_{z'}) \bullet \\
& \left. (e_{x'}^T \mathcal{R}^{\text{origin}} e_{y'}) \bullet (e_{x'}^T \mathcal{R}^{\text{origin}} e_{z'})) \right)) \quad (18)
\end{aligned}$$

5.3 First-Last to Even-Odd mapping

In this subsection we show that in contrast to the previous example, the origin graph of First-Last to Even-Odd mapping does not satisfy the condition of **k-crossing**. Patterns of this type are unattested linguistically, though they are reminiscent of certain types of spreading patterns.

Consider the origin transduction exemplified in fig. 4. Every odd position in the output has the last input position as its origin. Every even output position has the first input position as its origin. The length of the output can be arbitrary. All input positions are then sandwiched between the origins of each neighboring even-odd output position pair.

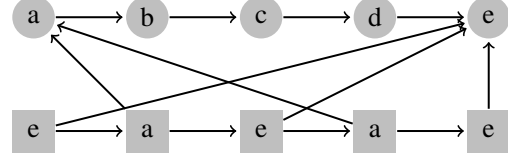


Figure 4: Visualizations of the origin graph of reversing *abcde*

And thus these input positions are crossed by each even position in the output. Suppose the length of the output is n . The crossing number is $\lfloor n/2 \rfloor$, which grows with n and is unbounded.

The essential property of origin graphs of first-last to even-odd mapping can be defined by the following MSO formula:

$$\begin{aligned}
F_{\text{FLEO}} = & \\
& \exists Y_{\text{output-o}}, Y_{\text{output-e}} (R_{\text{output-o/e}}(Y_{\text{output-o}}, Y_{\text{output-e}}) \\
& \wedge \forall y (y \in Y_{\text{output-o}} \rightarrow \exists x_f (R_{\text{first-input}}(x_f) \\
& \wedge R_{\text{origin}}(x_f, y)) \wedge (y \in Y_{\text{output-e}} \rightarrow \\
& \exists x_l (R_{\text{last-input}}(x_l) \wedge R_{\text{origin}}(x_l, y)))) \quad (19)
\end{aligned}$$

The MSO definable binary predicate $R_{\text{output-o/e}}(Y_{\text{output-o}}, Y_{\text{output-e}})$ is true when $Y_{\text{output-o}}$ and $Y_{\text{output-e}}$ constitute a partition of the set of output positions and $Y_{\text{output-o}}$ is the set of all odd output positions while $Y_{\text{output-e}}$ denotes the set of all even output positions (see e.g. Filiot, 2015). The formula asserts that every even output position has the last input position as its origin while every odd output position has the first input position as its origin position. The part requires they bear the same label is omitted for the ease of understanding.

We can convert this formula into prenex normal form as follows:

$$\begin{aligned}
& \exists Y_{\text{output-o}} \exists Y_{\text{output-e}} \forall y \exists x_f \exists x_l (\\
& R_{\text{output-o/e}}(Y_{\text{output-o}}, Y_{\text{output-e}}) \wedge \\
& \wedge (\neg y \in Y_{\text{output-o}} \vee (R_{\text{first-input}}(x_f) \wedge R_{\text{origin}}(x_f, y)) \wedge \\
& \wedge (\neg y \in Y_{\text{output-e}} \vee (R_{\text{last-input}}(x_l) \wedge R_{\text{origin}}(x_l, y)))) \quad (20)
\end{aligned}$$

We can compile this into a tensor formula as follows:

$$\begin{aligned}
\mathcal{T}_{\text{FLEO}} = & \min_1 \sum_{Y_{\text{output-o}}, Y_{\text{output-e}} \subseteq D} (1 - \min_1 \sum_{y=1}^N (1 - \\
& - \min_1 \sum_{x_f, x_l=1}^N (\mathcal{R}^{\text{output-o/e}} \times \mathbf{e}_{Y_{\text{output-o}}} \times \mathbf{e}_{Y_{\text{output-e}}}) \bullet \\
& \bullet \min_1 ((1 - \mathbf{e}_y \bullet \mathbf{e}_{Y_{\text{output-o}}}) + (\mathcal{R}^{\text{first-input}} \mathbf{e}_{x_f}) \bullet \\
& \bullet (\mathbf{e}_{x_f}^T \mathcal{R}^{\text{origin}} \mathbf{e}_y)) \bullet \min_1 ((1 - \mathbf{e}_y \bullet \mathbf{e}_{Y_{\text{output-e}}}) + \\
& + (\mathcal{R}^{\text{last-input}} \mathbf{e}_{x_l}) \bullet (\mathbf{e}_{x_l}^T \mathcal{R}^{\text{origin}} \mathbf{e}_y))) \quad (21)
\end{aligned}$$

6 Conclusion

This paper showed how to embed transductions in a vector space via operations over tensors. In particular, by using the idea of origin graphs, which represent input-output relations computed by some transducer, we embedded these graphs into tensors via finite model theory, and introduced Monadic Second-Order logical operations to compile the connectives and quantifiers. We showed how a class of origin graphs with these properties characterizing the regular transductions fits this exactly, and gave several examples motivated from linguistics.

There are several further directions this work could take. The most obvious is to consider the class of First-Order transductions on its own term. First-Order functions generalize the star-free languages (definable in first-order logic) to transductions, and correspond to restricting the underlying automaton of the transducer to be aperiodic (see [Filiot et al. \(2019\)](#)).

Transductions have also been extended to other structures besides strings, such as trees, which are relevant data structures in syntactic and semantic phenomena. The concept of origin information can be extended from string transducers to tree transducers, by considering the input and output graphs as tree structures ordered by dominance ([Filiot et al., 2018](#); [Winter, 2021](#)). Therefore, tree transductions can be embedded into vector space using the same methods.

In general, the flexibility given by model theory, as well as the precision given by classes of transductions, allows for multiple characterizations of structures of interest to linguistics, computer science, and cognitive science.

References

- Diederik Aerts. 2009. Quantum structure in cognition. *Journal of Mathematical Psychology*, 53(5):314–348.
- Rajeev Alur and Pavol Černý. 2010. Expressiveness of streaming string transducers. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, Leibniz International Proceedings in Informatics, pages 1–12, Germany. Dagstuhl Publishing.
- Peter Beim Graben, Dimitris Pinotsis, Douglas Saddy, and Roland Potthast. 2008. Language processing with dynamic fields. *Cognitive Neurodynamics*, 2(2):79–88.
- Reinhard Blutner. 2009. Concepts and bounded rationality: An application of niestegge’s approach to conditional quantum probabilities. In *AIP Conference Proceedings*, volume 1101, pages 302–310. AIP.
- Mikołaj Bojańczyk. 2014. Transducers with origin information. In *Automata, Languages, and Programming: 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II 41*, pages 26–37. Springer.
- Mikołaj Bojańczyk, Laure Daviaud, Bruno Guillon, and Vincent Penelle. 2017. Which classes of origin graphs are generated by transducers? In *ICALP 2017*.
- Nicolas Bourbaki. 1989. *Commutative Algebra: Chapters 1-7*. Springer-Verlag.
- J. Richard Büchi. 1960. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92.
- Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126(1):53–75.
- Bruno Courcelle and Joost Engelfriet. 2012. *Graph structure and monadic second-order logic: a language-theoretic approach*, volume 138. Cambridge University Press.
- Hossep Dolatian, Jonathan Rawski, and Jeffrey Heinz. 2021. Strong generative capacity of morphological processes. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 228–243.
- Herbert B. Enderton. 2001. *A Mathematical Introduction to Logic*, 2nd edition. Academic Press.
- Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic (TOCL)*, 2(2):216–254.
- Emmanuel Filiot. 2015. Logic-automata connections for transformations. In *Indian Conference on Logic and Its Applications*, pages 30–57. Springer.

- Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. 2019. Logical and algebraic characterizations of rational transductions. *Logical methods in computer science*, 15.
- Emmanuel Filiot, Sebastian Maneth, Pierre-Alain Reynier, and Jean-Marc Talbot. 2018. Decision problems of tree transducers with origin. *Information and Computation*, 261:311–335.
- Peter Gardenfors. 2004. Conceptual spaces as a framework for knowledge representation. *Mind and Matter*, 2(2):9–27.
- Thomas Graf. 2010. Logics of phonological reasoning. Master’s thesis, University of California, Los Angeles.
- Edward Grefenstette. 2013. Towards a formal distributional semantics: Simulating logical calculi with tensors. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*.
- Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frans Plank, editors, *Phonological Typology, Phonetics and Phonology*, chapter 5, pages 126–195. De Gruyter Mouton.
- Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.
- Leonid Libkin. 2004. *Elements of Finite Model Theory*. Springer.
- R Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. 2018. Rnns implicitly implement tensor product representations. *arXiv preprint arXiv:1812.08718*.
- Eric Meinhardt, Anna Mai, Eric Baković, and Adam McCollum. 2024. Weak determinism and the computational consequences of interaction. *Natural Language & Linguistic Theory*, pages 1–42.
- Phillip H. Miller. 1999. *Strong Generative Capacity: The Semantics of Linguistic Formalism*. Stanford, CA: CSLI Publications.
- Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.
- Max Nelson, Hossep Dolatian, Jonathan Rawski, and Brandon Prickett. 2020. Probing rnn encoder-decoder generalization of subregular functions using reduplication. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 167–178.
- Amanda Payne, Mai Ha Vu, and Jeffrey Heinz. 2016. A formal analysis of correspondence theory. In *Proceedings of the Annual Meetings on Phonology*.
- Geoffrey K. Pullum. 2007. The evolution of model-theoretic frameworks in linguistics. In *Model-Theoretic Syntax at 10*, pages 1–10, Dublin, Ireland.
- Stephan Rabanser, Oleksandr Shchur, and Stephan Günemann. 2017. Introduction to tensor decompositions and their applications in machine learning. *arXiv preprint arXiv:1711.10781*.
- Jonathan Rawski. 2019. Tensor product representations of subregular formal languages. In *Proceedings of the International Joint Conference on Artificial Intelligence workshop on Neural-Symbolic Learning and Reasoning*, pages 36–42.
- 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*. Oxford University Press, Oxford.
- James Rogers. 1998. *A descriptive approach to language-theoretic complexity*. CSLI Publications Stanford, CA.
- Taisuke Sato. 2017. Embedding tarskian semantics in vector spaces. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.
- Luciano Serafini and Artur S d’Avila Garcez. 2016. Learning and reasoning with logic tensor networks. In *Conference of the Italian Association for Artificial Intelligence*, pages 334–348. Springer.
- Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216.
- Paul Smolensky and Géraldine Legendre. 2006. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*, volume Volume I: Cognitive Architecture. MIT Press.
- Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. 2023. Transformers as recognizers of formal languages: A survey on expressivity. *arXiv preprint arXiv:2311.00208*.
- Whitney Tabor. 2009. A dynamical systems perspective on the relationship between symbolic and non-symbolic computation. *Cognitive neurodynamics*, 3(4):415–427.
- Wolfgang Thomas. 1997. Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, chapter 7. Springer.
- Sarah Winter. 2021. Decision problems for origin-close top-down tree transducers. In *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Can Syntactic Log-Odds Ratio Predict Acceptability and Satiation?

Jiayi Lu*, Jonathan Merchan*, Lian Wang*, Judith Degen

Department of Linguistics, Stanford University
{jiayi.lu, jmerchan, lianwang, jdegen}@stanford.edu

Abstract

The syntactic log-odds ratio (SLOR), a surprisal-based measure estimated from pre-trained language models (LMs) has been proposed as a linking function for human sentence acceptability judgments, a widespread measure of linguistic knowledge in experimental linguistics. We test this proposal in three steps: by examining whether SLOR values estimated by GPT-2 Small predict human acceptability judgments; by asking whether satiation effects observed in human judgments are also exhibited by fine-tuned LMs; and by testing whether satiation effects generalize in qualitatively similar ways in the model compared to humans. We show that SLOR in general predicts acceptability, but there is a significant amount of variance in acceptability data that SLOR fails to capture. SLOR also fails to capture certain patterns of satiation and generalization. Our results challenge the idea that surprisal alone, via a SLOR linking function, constitutes an accurate model for human acceptability judgments.

1 Introduction

Judgments of a sentence’s acceptability are commonly interpreted as a reflection of linguistic knowledge. For example, native English speakers find sentences like **What did John hear the rumor that Mary ate?* much less acceptable than sentences like *What did John hear that Mary ate?*. These kinds of acceptability judgments by native speakers have been widely used to inform linguistic theories. For example, based on the acceptability contrast in the aforementioned two sentences, linguists have proposed syntactic constraints (in this case, the “complex-NP island constraint”) to rule out the first sentence as ungrammatical (Ross, 1967).

Despite the widespread use of acceptability judgments as a source of evidence to inform linguistic theories, the cognitive mechanisms involved in

generating these judgments are rather poorly understood (Schütze, 1996; Sprouse, 2018; Francis, 2022). Past linguistic research has identified various factors that affect a sentence’s acceptability, including but not limited to its grammaticality, the frequency of observed lexical items and structures, task-related factors such as presentation order, and subject-related factors such as literacy and prior linguistic training (Schütze, 1996). However, there is no clearly spelled-out model that captures how these factors interact to give rise to an acceptability judgment. More recently, some studies hypothesized that there is a “surprisal bottleneck” for acceptability judgments: just as surprisal serves as a causal bottleneck for online processing difficulty (Levy, 2008), surprisal may also be the causal bottleneck for sentence acceptability (Lau et al., 2017, 2020; Culicover et al., 2022). If pre-trained language models (LMs) capture human linguistic knowledge, some studies argue that surprisal-based metrics estimated by LMs may serve as linking functions for human sentence acceptability judgments (Lau et al., 2017, 2020). In one prominent study, human sentence acceptability judgments were found to be best predicted by the syntactic log-odds ratio (SLOR, shown in Equation 1) values, a sentence’s model-given log probability normalized by its length and its probability based on its lexical items’ unigram probabilities (Lau et al., 2017):¹

$$\text{SLOR} = \frac{\log p_m(s) - \sum_{w \in s} \log p_u(w)}{|s|} \quad (1)$$

Here, $p_m(s)$ is the probability of a sentence s as estimated by the model (calculated as the product of the model-estimated transition probability for each word), $p_u(w)$ is the unigram probability of a word w in s , and $|s|$ is the sentence’s length in words. SLOR achieved the best correlation with

¹SLOR was first proposed by Pauls and Klein (2012) for a different task.

*These authors contributed equally.

sentence acceptability ratings among a variety of surprisal-based metrics.

In the present study, we revisit the hypothesis that SLOR estimates from pre-trained LMs provide a good linking function for acceptability judgments. We do so in three ways: first, we replicate the correlation between SLOR and sentence acceptability ratings using a more up-to-date LM than that used by Lau et al. (2017). Second, we move beyond one-shot acceptability ratings and investigate whether the changes in SLOR after fine-tuning predict the changes in human acceptability judgments in response to exposure (i.e. the “satiation effect”: Snyder, 2000; Chaves and Dery, 2019; Lu et al., 2021, inter alia). Third, we test whether fine-tuning the model with one sentence type leads to SLOR increase in a different but structurally related sentence type, replicating the generalization of satiation effects found in human acceptability judgment data (Lu et al., 2022).²

If the pre-trained LM approximates human linguistic knowledge and its SLOR estimates constitute a good linking function for human sentence acceptability judgments, SLOR values should correlate with acceptability judgments and demonstrate both human-like satiation effects and the generalization of satiation effects – both of which are phenomena that have been shown to reliably emerge in human acceptability judgment tasks (Snyder, 2000; Chaves and Dery, 2019; Lu et al., 2021, 2022).

2 Experiment 1: SLOR Predicts Acceptability

Experiment 1 aims to replicate Lau et al. (2017)’s finding that SLOR predicts sentence acceptability judgments using GPT-2 Small. We chose GPT-2 Small as opposed to other larger pre-trained LMs because GPT-2 Small’s surprisal estimates have been shown to best predict human reading time data among the GPT family (Oh and Schuler, 2023), suggesting that it is a more plausible candidate for a model that captures human linguistic knowledge than its relatives. Furthermore, it has been shown that GPT-2 demonstrates more human-like performance in forced-choice judgment tasks with minimal pair sentences involving island violations than other LLMs such as LSTM and Transformer-XL (Warstadt et al., 2020).³

²All datasets and scripts can be found here: <https://github.com/jmerch/slor-acceptability-satiation>.

³In Lau et al. (2017), the models tested were 2/3/4-gram models, BHMM, 2-tier BHMM, LDAHMM, and RNNLM,

2.1 Method and Procedure

We obtained the SLOR values for a wide range of sentences selected from recent studies that reported human acceptability judgment results (examples shown in Table 1). All SLOR values were calculated based on the surprisal estimates for the test sentences from a pre-trained GPT-2 Small model (Radford et al., 2019).⁴ If GPT-2 Small indeed captures human linguistic knowledge, and if the SLOR values estimated by LMs constitute a good linking function for sentence acceptability judgments as suggested in previous studies, the computed SLOR values should predict the acceptability judgments from the human experiments.

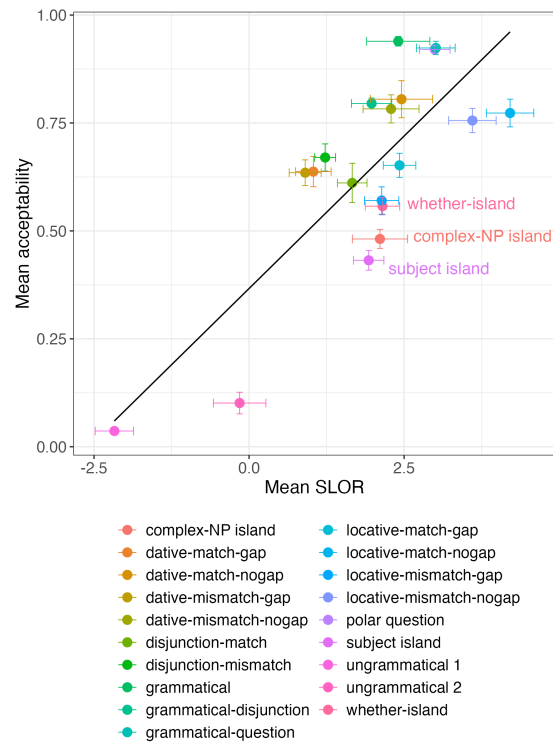


Figure 1: Plot of human acceptability judgments against model SLOR values. Error bars represent 95% bootstrapped confidence intervals. Points representing the three sentence types used as critical conditions in Experiments 2 and 3 (Complex-NP island, subject island, and *whether-island*) are labelled with text.

2.2 Results and Discussion

For the purpose of analysis, all human acceptability judgments from the original studies were linearly pre-trained on the BNC corpus and the English Wikipedia.

⁴We used the implementation of the 124M-parameter GPT-2 model from the *Transformers* library released by Hugging-Face (Wolf et al., 2019).

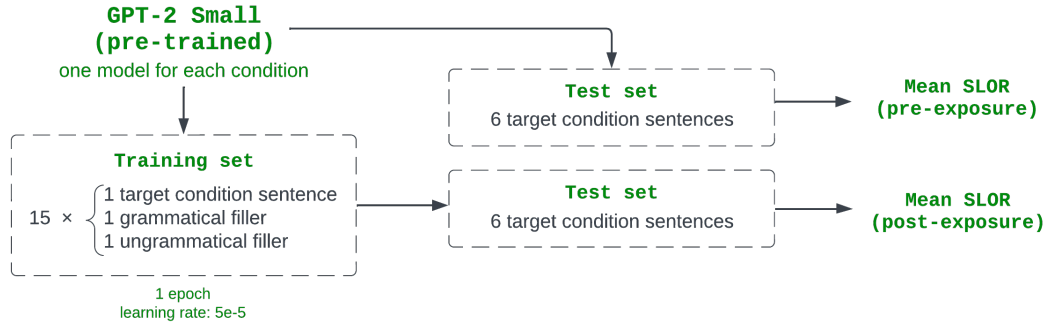


Figure 2: Experimental design of Experiment 2a

Condition	Example	Source
polar question	Does the teacher think that the boy found a box of diamonds?	
whether-island	What does the tourist wonder whether the lion attacked ___?	
subject island	What does the janitor think a bottle of ___ can remove the stain?	Lu et al. (2021, 2022)
complex-NP island	Who does the king believe the claim that the prince envied ___?	
grammatical-question	Who thinks that the doctor decided to treat the mysterious condition?	
ungrammatical 1	Who inspection did not restaurant pass health believes the claim that?	
dative-match-nogap	Kevin gave the children toys and Maria gave the teachers books.	
dative-match-gap	Kevin gave the children toys and Maria ___ the teachers books.	
dative-mismatch-nogap	Kevin gave the children toys and Maria gave books to the teachers.	
dative-mismatch-gap	Kevin gave the children toys and Maria ___ books to the teachers.	Lu and Kim (2022)
locative-match-nogap	Jacob brushed milk onto the pastry and Emily brushed oil onto the dough.	
locative-match-gap	Jacob brushed milk onto the pastry and Emily ___ oil onto the dough.	
locative-mismatch-nogap	Jacob brushed milk onto the pastry and Emily brushed the dough with oil.	
locative-mismatch-gap	Jacob brushed milk onto the pastry and Emily ___ the dough with oil.	
disjunction-match	Either Juan or Marie are making the decision.	
disjunction-mismatch	Either Juan or these teachers are making the decision.	
grammatical	Julia will perform the surgery tomorrow morning.	Lu and Degen (2023)
grammatical-disjunction	Either Juan or Marie is making the decision.	
ungrammatical 2	The scientists a discovered solution groundbreaking to	

Table 1: Example stimuli for each sentence type used in Exp. 1. Bolded types are critical conditions used in Exps. 2 and 3.

transformed to a value between 0 and 1 through min-max scaling, with 0 representing the “completely unacceptable” endpoint of the scale, and 1 representing the “completely acceptable” endpoint. Figure 1 shows the mean SLOR values against the mean human acceptability judgments for all the tested sentence types. In a linear regression, SLOR values significantly predicted the human judgments ($\beta = 0.080$, $SE = 0.005$, $t = 17.64$, $p < 0.001$), replicating the previous findings reported in Lau et al. (2017). The R^2 value of the model was 0.30, comparable to the best-performing model reported by Lau et al., an RNNLM as implemented by Mikolov (2012), trained on the English Wikipedia, and tested on a set of English Wikipedia sentences after round trip machine translation: $R^2 = 0.32$). The results suggest that the SLOR is a predictor of acceptability. However, we should also note that there is a significant amount of variance in the acceptability data that SLOR failed to capture,

challenging the hypothesis that the SLOR values estimated by the pre-trained GPT-2 Small constitute a full linking function for sentence acceptability.

3 Experiment 2a: Deriving Satiation Effects

One crucial property of human acceptability judgments is their malleability: ratings for initially degraded sentences tend to increase with repeated exposure. This effect has been called the “satiation effect” and has been reliably replicated in various sentence acceptability judgment studies (Snyder, 2000; Hiramatsu, 2001; Francom, 2009; Crawford, 2012; Chaves and Dery, 2014, 2019; Brown et al., 2021; Lu et al., 2021, 2022). Crucially, not all sentence types are equally susceptible to satiation: it has been repeatedly observed that certain sentence types resist satiation, and among those that do satiate, satiation rates vary by sentence type (Snyder, 2022; Lu et al., 2023). For example, complex-NP

island sentences show a lower satiation rate than other island sentences, such as subject and *whether*-island sentences (examples shown in Table 1).

In Experiment 2a, we further test whether SLOR provides a good linking function for acceptability judgments in two ways: first, by examining whether SLOR values exhibit the satiation effect (like human acceptability judgments); and second, by investigating whether the varying rates of satiation of different sentence types are predicted by changes in SLOR values after fine-tuning. We follow van Schijndel and Linzen (2018) in using fine-tuning to induce change in surprisal-based metrics from LMs, though our study differs from theirs in that we are interested in the linking function from surprisal to acceptability judgments, rather than to reading times.

3.1 Method and Procedure

This experiment aims to replicate the satiation experiment reported by Lu et al. (2021) using GPT-2 Small. In Lu et al. (2021), human participants were asked to rate the acceptability of three different types of sentences that violated island constraints: complex-NP island sentences, subject island sentences, and *whether*-island sentences. The ratings for all three sentence types increased with increasing presentation order, thus demonstrating the satiation effect. The results from Lu et al. (2021)’s human experiment are shown in Figure 3a.

Importantly, the complex-NP island sentences showed a lower rate of satiation than the other two sentence types. Although it is unclear what makes the complex-NP island sentences satiate at a slower rate, this rate difference has been observed repeatedly and is unlikely to be an artifact of the design (Lu et al., 2022, 2023).

To simulate the repeated exposure in acceptability judgment experiments that gives rise to satiation effects, we fine-tuned GPT-2 Small models using the sentences from Lu et al. (2021). The schematic sketch of the experimental design is shown in Figure 2. For each of the three island types, we fine-tuned a GPT-2 Small model with 45 sentences from the human experiment, consisting of 15 grammatical fillers, 15 ungrammatical fillers, and 15 critical island sentences. The motivation for including the fillers in the training set was to simulate the human experimental experience as closely as possible.

3.2 Results and Discussion

Figure 3b shows the pre- and post-exposure SLOR values. The model-estimated post-exposure SLOR values were higher, by a factor of almost 3, than the pre-exposure values for all three sentence types. This suggests that GPT-2 demonstrates satiation-like behavior in response to exposure to degraded sentences. However, the relative ranking of satiation rates observed in the human results (Figure 3a) was not replicated: in the human experiment, complex-NP island sentences exhibited a significantly lower satiation rate than the other two sentence types; in contrast, the SLOR values for complex-NP sentences increased at a similar rate as *whether*-island sentences, which was higher than the subject island sentences. Thus, the change in SLOR values from by fine-tuning does not reflect the qualitative patterns of change in acceptability ratings through satiation beyond generally showing an increase. This poses a challenge to the proposal to treat SLOR values estimated from LMs as a full linking function for acceptability judgments.

However, there is a caveat to the interpretation of these results: the sentences used for fine-tuning and the sentences in the post-exposure test set contained considerable lexical overlap. In particular, all the complex-NP island sentences from Lu et al. (2021) contained the word sequence “... believe the claim that ...”. There was much less lexical overlap between training and test sentences in the other two conditions. It is thus possible that the large increase in SLOR for the complex-NP island condition was mostly driven by lexical repetition. To test this hypothesis, we adopt the same design as Experiment 2a in Experiment 2b but with a modified set of training sentences that controlled for lexical repetition.

4 Experiment 2b: Lexical Repetition Control

In this experiment, we test whether the model satiation pattern observed in Experiment 2a persists when we adopt a modified set of training sentences that control for lexical repetition.

4.1 Method and Procedure

The same design as Experiment 2a was adopted. The only difference was that the training set sentences were modified to maximally reduce the repetition of lexical items without changing the sentence’s syntactic structure. Whereas the complex-

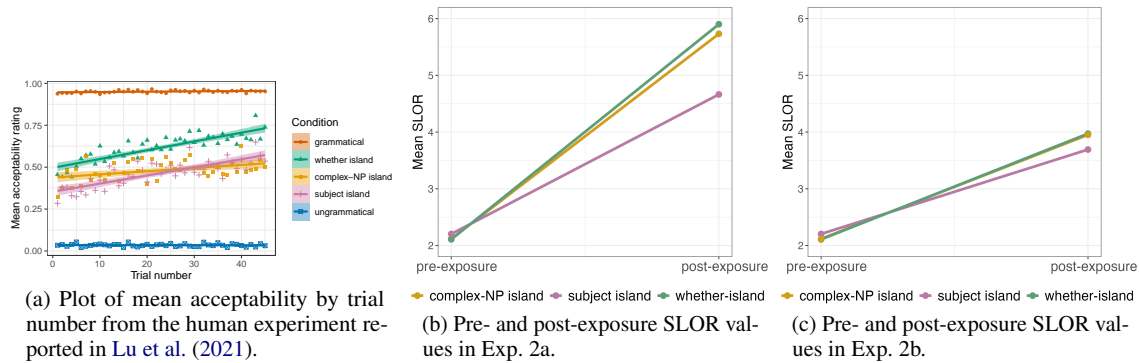


Figure 3: Comparison of human acceptability judgments reported in Lu et al. (2021) showing satiation effects (a), and model results from Exps. 2a and 2b (b-c).

NP sentences in the original training set all contained the word sequence “believe the claim that”, the complex-NP sentences in the modified training set all contained different matrix predicates. Similar modifications were also applied to the subject island sentences and the *whether*-island sentences to minimize lexical repetition (see the design schema in Figure 4).

4.2 Results and Discussion

The pre-exposure and post-exposure SLOR values for all three island sentence types are shown in Figure 3c. The SLOR values increased for all three sentence types post-exposure by about a factor of 2, i.e., at a lower rate than in Experiment 2a. This suggests that lexical repetition did indeed contribute to the large satiation rates observed in Experiment 2a. However, the relative order of the three sentence types’ SLOR increases remained the same as in Experiment 2a: the SLOR increase for the complex-NP sentences was comparable to that of the *whether*-island sentences, and higher than that of the subject island sentences. Thus, the comparatively lower satiation rate for complex-NP island sentences observed in the human results was once again not replicated.

In sum, the results from Experiments 2a and 2b demonstrate that GPT-2 Small exhibits satiation-like behavior with repeated exposure to degraded sentences. However, the magnitude and particular patterns of the SLOR increase do not mirror the human satiation effects. There are at least two potential explanations for this discrepancy. First, it is possible that the cognitive processes underlying the satiation effect observed in humans is qualitatively different from the fine-tuning process for LMs. Second, it is possible that the set of linguistic features that affect human satiation are different from the

ones that GPT-2’s surprisal estimate is sensitive to. Either way, these results challenge both the hypothesis that LM-derived SLOR estimates provide a full linking function for human sentence acceptability judgments, as well as the idea that GPT-2 Small fully captures human linguistic knowledge.

5 Experiment 3: Generalizing Satiation Effects

Another key property of human sentence acceptability judgments is that the acceptability increase gained through satiation generalizes across syntactically related sentence types (Lu et al., 2022). In a series of acceptability judgment experiments employing the same exposure-and-test paradigm as described above, Lu et al. (2022) exposed participants to one of three sentence types: subject island sentences, *whether*-island sentences, and polar questions. In the test phase, participants were asked to rate the acceptability of either subject island sentences or *whether*-island sentences. Exposure and test sentence types were fully crossed. The results are shown in Figures 6a and 6b. Conditions where participants were exposed to one island sentence type and tested on the other (e.g., exposed to subject island sentences and tested on *whether*-island sentences) are labeled “between-category”; conditions where participants were exposed to and tested on the same sentence type are labeled “within-category”. Acceptability ratings on test sentences were lower in the between-category than in the within-category condition, but significantly higher than in the control condition, where participants were exposed to polar questions (i.e., non-island sentences) and tested on island sentences. Lu et al. (2022) concluded from these results that the abstract linguistic features shared between the two

	Training set	Test set
Complex-NP island	Who does the detective state the hypothesis that a bottle of poison killed? Who does the bartender know the fact that the brother of the mayor invited? What does the president doubt the prediction that the senate will review?	What does the mechanic believe the claim that a tank of biofuel can power? Who do the activists believe the claim that government officials bribed? What does the musician believe the claim that the company will buy?
Subject island	What does the pianist believe that two hours of per day leads to perfection? What does the headmaster guess that an expert in wrote the manuscript? What do the delinquents say that another group of was arrested?	What does the doctor think that the proposal for was vetoed by the mayor? What did the pharmacist think that a pack of could cause nausea? Who does the pilot think that the description of matches the suspect?
Whether-island	What does the mechanic assess whether a tank of biofuel can power? What does the biologist doubt whether researchers will eventually find? Who do the delinquents discuss whether the police arrested?	What does the actor wonder whether the famous scholar wrote? What does the chef wonder whether the food critic will publish? What does the spy wonder whether the commander initiated?

Figure 4: Modified training and test sets used in Experiment 2b to control for lexical repetition

syntactically-related island sentence types (e.g., the existence of long-distance wh-movement, the existence of dependencies violating the subjacency condition, and others) can be used by participants as representational targets for satiation. The polar question sentences are less syntactically similar to the island sentences than the island sentences are to each other. As a result, when participants were exposed to polar questions in the exposure phase, there were fewer shared linguistic representations between the exposure and test sentences that could serve as representational targets for satiation, thus resulting in a smaller satiation generalization effect.

In this experiment, we adopted a similar design as Lu et al. (2022)’s human experiment with GPT-2 Small, with the aim to test whether the SLOR value estimates demonstrate the satiation generalization effect.

5.1 Method and Procedure

The schematic sketch of the experimental design is shown in Figure 5. We fine-tuned a pre-trained GPT-2 Small model with 12 exposure sentences (one of the three sentence types: subject island sentences, *whether*-island sentences, and polar question sentences) and 12 fillers in the training phase. In the test phase, we calculated the fine-tuned models’ SLOR estimates for two test sets consisting of subject island and *whether*-island sentences respectively. If the model demonstrates human-like satiation generalization effects, the post-exposure SLOR values should be higher than the pre-exposure values, the SLOR increase in the between-category condition should be smaller or equal to the SLOR increase in the within-category condition, and the SLOR increase in both the between- and within-category condition should be larger than in the control condition.

5.2 Results and Discussion

The results of Experiment 3 are shown in Figures 6c and 6d. In both test sets, the post-exposure SLOR values were higher than the pre-exposure SLOR values (indicated by the dashed lines in the figures) for all conditions. The SLOR increase for the between-category condition is numerically smaller than the within-category condition, similar to the pattern observed in the human results.

However, there was one unexpected observation. The SLOR increase for the control training condition (i.e., when the model was fine-tuned on polar questions and tested on either of the island sentence types) was comparable to the between-category condition when the model was tested on *whether*-island sentences, and even numerically larger than the between-category condition when the model was tested on subject island sentences. This suggests that for the model, the satiation generalization effect from polar questions to the island sentence types was comparable to, if not larger than, the satiation generalization between the two syntactically closely related island sentence types. By contrast, in the human results reported by Lu et al. (2022), the satiation generalization effect from polar questions to island sentences was the smallest among all training conditions.

In sum, we observed satiation generalization effects in the SLOR values estimated by GPT-2 Small. However, the control condition (i.e., the model fine-tuned on polar questions) showed an unexpectedly large satiation generalization effect that was even numerically larger than the between-category condition (at least when testing on subject island sentences). This suggests that the model treats the polar questions as more similar to the subject island sentences than the *whether*-island sentences.

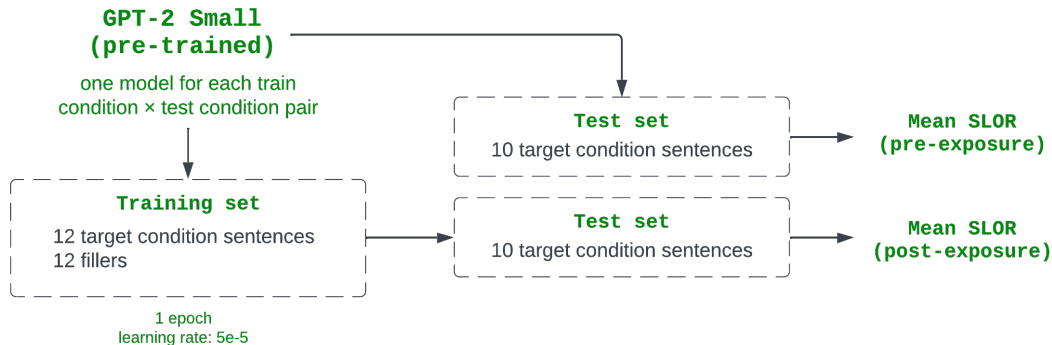


Figure 5: Experimental design of Experiment 3

By contrast, the human results suggest that there are more shared representations between the two island sentence types than between polar questions and either of the island sentence types. There are several possible explanations for this difference between the human and the model results: it is possible that the linguistic features that humans and the model pay attention to during fine-tuning/satiation are different; it is also possible that the three tested sentence types are represented in vastly different ways between humans and the model. Either way, these results again challenge both the hypothesis that LM-derived SLOR estimates provide a full linking function for human sentence acceptability judgments, as well as the idea that LMs fully capture human linguistic knowledge.

6 General Discussion

In this study, we aimed to test the hypothesis that SLOR values estimated by LMs can provide a linking function for human sentence acceptability judgments. We did so by testing pre-trained GPT-2 Small models in experiments following similar designs as various human sentence acceptability judgment studies, following the recent trend in the computational linguistic literature to treat LMs as subjects in experimental syntax and psycholinguistic experiments (Leong and Linzen, 2023; Futrell et al., 2018, 2019; Wilcox et al., 2023; Arehalli et al., 2022, *inter alia*). We compared the model performance against human results along three dimensions: (1) whether the model-estimated SLOR values predicted human acceptability judgments, (2) whether the increase in SLOR values through model fine-tuning exhibited the same qualitative patterns as the satiation patterns observed in human acceptability judgment experiments exposing participants to degraded sentences, and (3) whether the

increase in SLOR values through model fine-tuning exhibited the same qualitative generalization patterns across sentence types as observed in humans.

In Experiment 1, we showed that the SLOR values estimated by the pre-trained GPT-2 Small model predict sentence acceptability judgments given by human participants across a broad range of sentence types, replicating previous results that did not use Transformer models (Lau et al., 2017, 2020). This result suggests that the SLOR values estimated by GPT-2 Small is a plausible linking function for human acceptability judgments broadly. However, there was a lot of variance left unexplained by the SLOR values, suggesting that the linking function proposal is limited.

In Experiments 2a and 2b we showed that the SLOR values estimated by GPT-2 Small for degraded sentence types increase when the model is fine-tuned on sentences of the same structure, akin to the satiation effect observed in human participants. However, the magnitude of SLOR increase did not predict the magnitude of acceptability increase for the sentence types we tested. In Experiment 3, we further showed that models fine-tuned on one sentence type showed increased SLOR values for other sentence types, similar to the satiation generalization effect observed in human acceptability judgments experiments. However, the fine patterns of the generalization effect in the models was crucially different from the human results: fine-tuning on polar questions led to a greater SLOR increase for subject island sentences than fine-tuning on *whether*-island sentences, which are more syntactically similar to subject island sentences than polar questions.

In sum, we found that SLOR, a surprisal-based metric, generally predicts sentence acceptability. Fine-tuning LMs as a way of exposing them to

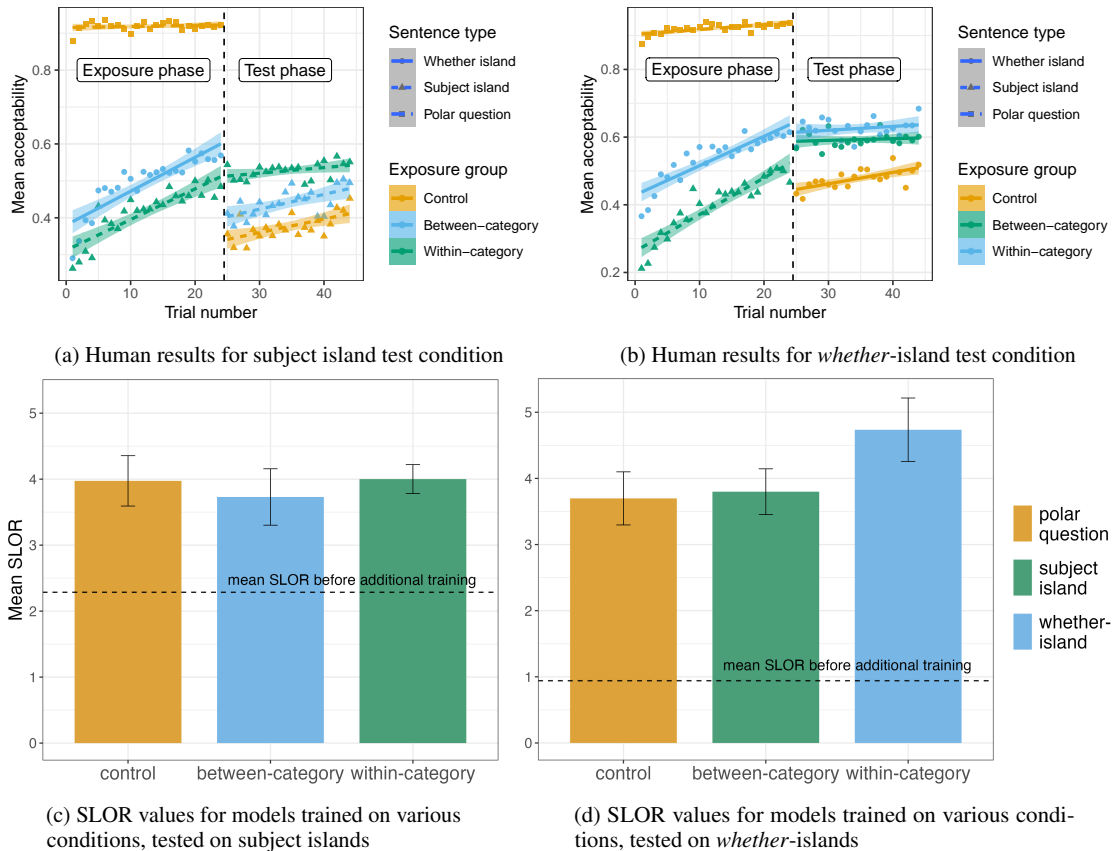


Figure 6: Comparison of the satiation generalization effect observed in the human experiments in Lu et al. (2022), shown in sub-figures (a-b), and the model results from Exp. 3, shown in sub-figures (c-d).

novel sentences leads to satiation and generalization effects, but the model results crucially differ from the human results in the fine patterns of the satiation and generalization effects. Our results suggest that LMs, like humans, are sensitive to abstract linguistic representations beyond lexical identity and particular sentence structures. However, the discrepancies with the human results highlight the differences in the relevant linguistic representations or the learning mechanisms between humans and language models, challenging the claim that pre-trained LMs like GPT-2 Small can fully capture human linguistic knowledge, or that SLOR estimated by such LMs can fully account for sentence acceptability judgments.

Finally, the results of the current study point to some possible directions for future research. Although we showed that SLOR estimated by GPT-2 does not fully capture human acceptability judgments, this does not definitively reject the hypothesis that surprisal is a causal bottleneck for acceptability (Lau et al., 2017, 2020; Culicover et al.,

2022). In order to further investigate the validity of the surprisal bottleneck hypothesis, future studies should examine LMs other than the ones we and the previous literature tested with the aim to gain surprisal estimates that more accurately capture human linguistic knowledge, and also examine metrics other than SLOR that may serve as better linking functions between surprisal and sentence acceptability.

References

- Suhas Arehalli, Brian Dillon, and Tal Linzen. 2022. Syntactic surprisal from neural models predicts, but underestimates, human processing difficulty from syntactic ambiguities. *Proceedings of the 26th Conference on Computational Natural Language Learning*, pages 301–313.
- Jessica MM Brown, Gisbert Fanselow, Rebecca Hall, and Reinhold Kliegl. 2021. Middle ratings rise regardless of grammatical construction: Testing syntactic variability in a repeated exposure paradigm. *PLOS One*, 16(5):e0251280.

- Rui P Chaves and Jeruen E Dery. 2014. Which subject islands will the acceptability of improve with repeated exposure. In *Proceedings of the 31st West Coast Conference on Formal Linguistics*, pages 96–106.
- Rui P Chaves and Jeruen E Dery. 2019. Frequency effects in subject islands. *Journal of Linguistics*, 55(3):475–521.
- Jean Crawford. 2012. Using syntactic satiation to investigate subject islands. In *Proceedings of the 29th West Coast Conference on Formal Linguistics*, pages 38–45. Cascadilla Proceedings Project Somerville, MA.
- Peter W Culicover, Giuseppe Varaschin, and Susanne Winkler. 2022. The radical unacceptability hypothesis: Accounting for unacceptability without universal constraints. *Languages*, 7(2):96.
- Elaine Francis. 2022. *Gradient acceptability and linguistic theory*, volume 11. Oxford University Press.
- Jerid Cole Francom. 2009. *Experimental Syntax: Exploring the effect of repeated exposure to anomalous syntactic structure—evidence from rating and reading tasks*. Ph.D. thesis, The University of Arizona.
- Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. Rnns as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.
- Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. *arXiv preprint arXiv:1903.03260*.
- Kazuko Hiramatsu. 2001. *Assessing linguistic competence: Evidence from children’s and adults’ acceptability judgments*. Ph.D. thesis, University of Connecticut.
- Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How furiously can colorless green ideas sleep? sentence acceptability in context. *Transactions of the Association for Computational Linguistics*, 8:296–310.
- 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.
- Cara Su-Yi Leong and Tal Linzen. 2023. [Language models can learn exceptions to syntactic rules](#). In *Proceedings of the Society for Computation in Linguistics 2023*, pages 133–144, Amherst, MA. Association for Computational Linguistics.
- Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.
- Jiayi Lu and Judith Degen. 2023. Acceptability judgment dataset for subject-verb agreement with disjoint subjects. *GitHub repository*.
- Jiayi Lu, Michael C Frank, and Judith Degen. 2023. [A meta-analysis of syntactic satiation in extraction from islands](#). *lingbuzz/007198*.
- Jiayi Lu and Nayoun Kim. 2022. The puzzle of argument structure mismatch in gapping. *Frontiers in Psychology*.
- Jiayi Lu, Daniel Lassiter, and Judith Degen. 2021. Syntactic satiation is driven by speaker-specific adaptation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43.
- Jiayi Lu, Nicholas Wright, and Judith Degen. 2022. Satiation effects generalize across island types. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.
- Tomáš Mikolov. 2012. *Statistical language models based on neural networks*. Ph.D. thesis, Brno University of Technology.
- Byung-Doh Oh and William Schuler. 2023. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350.
- Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- John R Ross. 1967. *Constraints on variables in syntax*. Ph.D. thesis, Massachusetts Institute of Technology.
- Carson T Schütze. 1996. *The Empirical Base of Linguistics: Grammaticality Judgments and Linguistic Methodology*. University of Chicago Press.
- William Snyder. 2000. An experimental investigation of syntactic satiation effects. *Linguistic Inquiry*, 31(3):575–582.
- William Snyder. 2022. Satiation. In Grant Goodall, editor, *The Cambridge Handbook of Experimental Syntax*, pages 154–180. Cambridge University Press.
- Jon Sprouse. 2018. Acceptability judgments and grammaticality, prospects and challenges. In *Syntactic structures after 60 years*, pages 195–224. De Gruyter Mouton.
- Martin van Schijndel and Tal Linzen. 2018. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mo-
hananey, Wei Peng, Sheng-Fu Wang, and Samuel R
Bowman. 2020. Blimp: The benchmark of linguistic
minimal pairs for english. *Transactions of the Asso-
ciation for Computational Linguistics*, 8:377–392.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy.
2023. Using computational models to test syntactic
learnability. *Linguistic Inquiry*, pages 1–44.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien
Chaumond, Clement Delangue, Anthony Moi, Pier-
ric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz,
and Jamie Brew. 2019. [Huggingface’s transformers:](#)
[State-of-the-art natural language processing.](#) *CoRR*.

Learning Phonotactics from Linguistic Informants

Canaan Breiss^{a,†} Alexis Ross^{b,†}

Amani Maina-Kilaas^b Roger Levy^b Jacob Andreas^b

^aUniversity of Southern California ^bMassachusetts Institute of Technology
cbreiss@usc.edu {alexisro, amanirmk, rplevy, jda}@mit.edu

Abstract

We propose an interactive approach to language learning that utilizes linguistic acceptability judgments from an informant (a competent language user) to learn a grammar. Given a grammar formalism and a framework for synthesizing data, our model iteratively selects or synthesizes a data-point according to one of a range of information-theoretic policies, asks the informant for a binary judgment, and updates its own parameters in preparation for the next query. We demonstrate the effectiveness of our model in the domain of phonotactics, the rules governing what kinds of sound-sequences are acceptable in a language, and carry out two experiments, one with typologically-natural linguistic data and another with a range of procedurally-generated languages. We find that the information-theoretic policies that our model uses to select items to query the informant achieve sample efficiency comparable to, and sometimes greater than, fully supervised approaches.

1 Introduction

In recent years, natural language processing has made remarkable progress toward models that can (explicitly or implicitly) predict and use representations of linguistic structure from phonetics to syntax (Mohamed et al., 2022; Hewitt and Manning, 2019). These models play a prominent role in contemporary computational linguistics research. But the data required to train them is of a vastly larger scale, and features less controlled coverage of important phenomena, than data gathered in the course of linguistic research, e.g. during language documentation with native speaker informants. How can we build computational models that learn more *like linguists*—from targeted inquiry rather than large-scale corpus data?

We describe a paradigm in which language-learning agents interactively select examples to

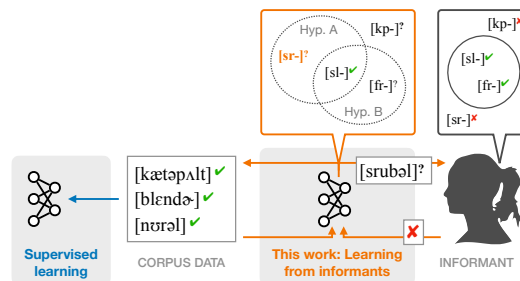


Figure 1: Overview of our approach. Instead of learning a model from a static set of well-formed word forms (left), we interactively elicit acceptability judgments from a knowledgeable language user (right), using ideas from active learning and optimal experiment design. On a family of phonotactic grammar learning problems, active example selection is sometimes more sample-efficient than supervised learning or elicitation of judgments about random word forms.

learn from by querying an **informant**, with the goal of learning about a language as data-efficiently as possible, rather than relying on large-scale corpora to capture attested-but-rare phenomena. This approach has two important features. First, rather than relying on existing data to learn, our model performs **data synthesis** to explore the space of useful possible data-points. But second, our model can also **leverage corpus data** as part of its learning procedure by trading off between interactive elicitation and ordinary supervised learning, making it useful both *ab initio* and in scenarios where seed data is available to bootstrap a full grammar.

We evaluate the capabilities of our methods in two experiments on learning *phonotactic grammars*, in which the goal is to learn the constraints on sequences of permissible sounds in the words of a language. Applied to the problem of learning a vowel harmony system inspired by natural language typology, we show that our approach succeeds in recovering the generalizations governing the distribution of vowels. Using an ad-

[†]Both authors contributed equally to this work.

ditional set of procedurally-generated synthetic languages, our approach also succeeds in recovering relevant phonotactic generalizations, demonstrating that model performance is robust to whether the target pattern is typologically common or not. We find that our approach is more sample-efficient than ordinary supervised learning or random queries to the informant.

Our methods have the potential to be deployed as an aid to learners acquiring a second language or to linguists doing elicitation work with speakers of a language that has not previously been documented. Further, the development of more data-efficient computational models can help redress social inequalities which flow from the asymmetrical distribution of training data types available for present models (Bender et al., 2021).

2 Problem Formulation and Method

Preliminaries We aim to learn a language L comprising a set of strings x , each of which is a concatenation of symbols from some inventory Σ (so $L \subseteq \Sigma^+$). (In phonotactics, for example, Σ might be the set of phonemes, and L the set of word forms that speakers judge phonotactically acceptable.) A learned model of a language is a discriminative function that maps from elements $x \in \Sigma^+$ to values in $\{0, 1\}$ where 1 indicates that $x \in L$ and 0 indicates that $x \notin L$. In this paper, we will generalize this to **graded** models of language membership $f : \Sigma^+ \mapsto [0, 1]$, in which higher values assigned to strings $x \in \Sigma^+$ correspond to greater confidence that $x \in L$ (cf. Albright, 2009, for data and argumentation in favor of a gradient model of phonotactic acceptability in humans).

We may then characterize the language learning problem as one of acquiring a collection of pairs $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ where $x_i \in \Sigma^+$, and $y_i \in \{0, 1\}$ correspond to **acceptability judgments** about whether $x_i \in L$. Given this data, a learner’s job is to identify a language consistent with these pairs. Importantly, in this setting, learners may have access to both positive and negative evidence.

Approach In our problem characterization, the data acquisition process takes place over a series of time steps. At each time step t , the learner uses a **policy** π according to which a new string $x_t \in \mathcal{X}$ is selected; here \mathcal{X} is some set of possible strings, with $L \subset \mathcal{X} \subset \Sigma^+$. The chosen string is then passed to an **informant** that provides the learner

Algorithm 1: Iterative Query Procedure

Input: policy π , total timesteps T
 $(\underline{x}, \underline{y}) \leftarrow []; t \leftarrow 0;$
while $t < T$ **do**
 $x_t \leftarrow \pi(x \mid \underline{x}, \underline{y});$
 $y_t \leftarrow \text{informant}(x_t);$
 append (x_t, y_t) to $(\underline{x}, \underline{y});$
 $t \leftarrow t + 1;$
end

a value $y_t \in \{0, 1\}$ corresponding to whether x_t is in L . The new datum (x_t, y_t) is then appended to a running collection of (string, judgment) pairs $(\underline{x}, \underline{y})$, after which the learning process proceeds to the next time step. This procedure is summarized in Algorithm 1.

Conceptually, there are two ways in which a learner might gather information about a new language. One possibility is to gather examples well-formed strings already produced by users of the language (e.g. by listening to a conversation, or collecting a text corpus), similar to an “immersion” approach when learning a new language. In this case, the learner does not have control over the specific selected string x_t , but it is guaranteed that the selected string is part of the language: $x_t \in L$ and thus $y_t = 1$.

The other way of collecting information is to select some string x_t from \mathcal{X} , and directly elicit a judgment y_i from a knowledgeable informant. This approach is often pursued by linguists working with language informants in a documentation setting, where their query stems from a hypothesis about the structural principles of the language. Here, examples can be chosen to be maximally informative, and negative evidence gathered directly. In practice, learners might also use “hybrid policies” that compare which of multiple basic policies (passive observation, active inquiry) is expected to yield a new datum that optimally improves the learner’s knowledge state. Each of these strategies is described in more detail below.

Model assumptions To characterize the learning policies, we make the following assumptions regarding the **model** trained from available data $(\underline{x}, \underline{y})$. We assume that the function $f : \Sigma^+ \rightarrow [0, 1]$ acquired from $(\underline{x}, \underline{y})$ can be interpreted as a conditional probability of the form $p(y \mid x, \underline{x}, \underline{y})$. We further assume that this conditional probability is determined by a set of parameters θ

for which a(n approximate) posterior distribution $P(\theta \mid \underline{x}, \underline{y})$ is maintained, with $p(y \mid x, \underline{x}, \underline{y}) = \int_{\theta} P(y|x, \theta)P(\theta \mid \underline{x}, \underline{y}) d\theta$.

3 Query policies

In the framework described in Section 2, how should a learner choose which questions to ask the informant? Below, we describe a family of different policies for learning.

3.1 Basic policies

Train The first basic policy, $\pi_{\text{train}}(x \mid \underline{x}, \underline{y})$, corresponds to observing and recording an utterance by a speaker. For simplicity we model this as uniform sampling (without replacement) over L :

$$\pi_{\text{train}}(x \mid \underline{x}, \underline{y}) \sim U(\{x \in L - \underline{x}\}).$$

Uniform The second basic policy, $\pi_{\text{unif}}(x \mid \underline{x}, \underline{y})$, samples a string uniformly from \mathcal{X} and presents it to the informant for an acceptability judgment:

$$\pi_{\text{unif}}(x \mid \underline{x}, \underline{y}) \sim U(\{x \in \mathcal{X}\}).$$

Label Entropy The $\pi_{\text{label-ent}}(x \mid \underline{x}, \underline{y})$ policy selects the string x^* with the maximum entropy \mathcal{H} over labels y under the current model state:

$$x^* = \arg \max_{x \in \mathcal{X}} \mathcal{H}(y \mid x, \underline{x}, \underline{y}).$$

Expected Information Gain The $\pi_{\text{eig}}(x \mid \underline{x}, \underline{y})$ policy selects the candidate that, if observed, would yield the greatest expected reduction in entropy over the posterior distribution of the model parameters θ . This is often called the **information gain** (MacKay, 1992); we denote the change in entropy as $V_{\text{IG}}(x, y; \underline{x}, \underline{y})$:

$$\begin{aligned} V_{\text{IG}}(x, y; \underline{x}, \underline{y}) \\ = \mathcal{H}(\theta \mid \underline{x}, \underline{y}) - \mathcal{H}(\theta \mid x, y, \underline{x}, \underline{y}). \end{aligned} \quad (1)$$

The expected information gain policy selects the x^* that maximizes $\mathbb{E}_{y \in [0,1]} V_{\text{IG}}(x, y; \underline{x}, \underline{y})$, i.e.:

$$\begin{aligned} x^* = \arg \max_{x \in \mathcal{X}} \\ V_{\text{IG}}(x, y = 1; \underline{x}, \underline{y}) \cdot p(y = 1 \mid x, \underline{x}, \underline{y}) \\ + V_{\text{IG}}(x, y = 0; \underline{x}, \underline{y}) \cdot p(y = 0 \mid x, \underline{x}, \underline{y}), \\ \pi_{\text{eig}}(x \mid \underline{x}, \underline{y}) = \delta(x^*), \end{aligned}$$

where $\delta(x)$ denotes the probability distribution that places all its mass on x .

3.2 Hybrid Policies

Hybrid policies dynamically choose at each time step among a set of basic policies Π based on some metric V . At each step, the hybrid policy estimates the expected value of V for each basic policy $\pi \in \Pi$, chooses the policy π^* that has the highest expected value, and then samples $x \in \Sigma^+$ according to π^* . Here, we study one such policy: $\Pi = [\pi_{\text{train}}, \pi_{\text{eig}}]$, with metric $V = V_{\text{IG}}$. We refer to the non-train policy as $\hat{\pi}$ and the metric used to select $\pi^* \in [\hat{\pi}, \pi_{\text{train}}]$ at each step as V .

We explore two general methods for estimating the expected value of V for each policy π^* : *history-based* and *model-based*. We also explore a *mixed* approach using a history-based method for π_{train} and a model-based method for $\hat{\pi}$.

History In the history-based approach, the model keeps a running average of empirical values of V for candidates previously selected by π_{train} and $\hat{\pi}$.

For instance, for history-based hybrid policy $\pi_{\text{eig-history}}(x \mid \underline{x}, \underline{y})$, $V = V_{\text{IG}}$ (see Table 1b). Suppose at a particular step, the basic policy π^* selected by $\pi_{\text{eig-history}}$ chose query x , which received label y from the informant. Then the history-based $\pi_{\text{eig-history}}$ would store the empirical information gain between $p(\theta \mid \underline{x}, \underline{y})$, $p(\theta \mid x, y, \underline{x}, \underline{y})$ for the chosen π^* ; in future steps, it would then select the $\pi^* \in [\pi_{\text{train}}, \hat{\pi}]$ with the highest empirical mean of V , in this case the empirical mean information gain, over candidates queried by each basic policy.

More formally, let $S^{\text{EMP}}(\pi; \underline{x}, \underline{y})$ refer to the mean of observed values V for candidates x_i selected by π before step t , where $\pi \in [\pi_{\text{train}}, \hat{\pi}]$:

$$S^{\text{EMP}}(\pi; \underline{x}, \underline{y}) = \frac{\sum_{i \in I_{\pi}} V(x_i, y_i; \underline{x}_{<i}, \underline{y}_{<i})}{|I_{\pi}|},$$

where $I_{\pi} = \{i \mid x_i \text{ was selected by } \pi, i < t\}$.

$V(x_i, y_i; \underline{x}_{<i}, \underline{y}_{<i})$ denotes V 's score for the i 'th datapoint x_i selected by π under a model that as observed data $\underline{x}_{<i}, \underline{y}_{<i}$.

Then at step t , the history-based hybrid policies sample π^* according to:

$$\pi^* = \arg \max_{\pi \in [\hat{\pi}, \pi_{\text{train}}]} S^{\text{EMP}}(\pi; \underline{x}, \underline{y}).$$

For $t < 2$, we automatically select π_{train} and $\hat{\pi}$ in a random order, each once, to ensure we have empirical means for both policies.

Model The model-based approach is prospective and involves using the current posterior distribution over model parameters to compute an expected value for the target metric under the policy. We define two ways of computing these expectations.

$S^{\text{EXP}(y)}$ computes an expectation over possible **labels** y for the candidate x^* that will be chosen by policy π . We use $S^{\text{EXP}(y)}$ to score **non-train** basic policies $\hat{\pi}$ because they select x^* deterministically given \mathcal{X} , *i.e.*, selecting the inputs that maximize the objectives described in §3.1. More formally:

$$S^{\text{EXP}(y)}(\hat{\pi}; \underline{x}, \underline{y}) = \mathbb{E}_{y \in [0,1]} V(x^*, y; \underline{x}, \underline{y}), x^* \sim \hat{\pi}.$$

$S^{\text{EXP}(x)}$ computes an expectation over possible **inputs** $x \in L$ and assumes a fixed label ($y = 1$). We score the **train** basic policy π_{train} with $S^{\text{EXP}(x)}$ because the randomness for π_{train} is over forms in the lexicon that could be sampled by π_{train} , and labels are always 1. More formally:

$$S^{\text{EXP}(x)}(\pi_{\text{train}}; \underline{x}, \underline{y}) = \mathbb{E}_{x \in L} V(x, y = 1; \underline{x}, \underline{y}).$$

In practice, however, we approximate this expectation with samples from \mathcal{X} , since we do not assume that the model has access to the lexicon used by the informant. In particular, we model the probability that a form x is in the lexicon as $p(y = 1 \mid x; \underline{x}, \underline{y})$.

Using the policy-specific expectations defined above, the model-based approach selects the policy π^* according to:

$$\pi^* = \arg \max_{\pi \in [\hat{\pi}, \pi_{\text{train}}]} S(\pi; \underline{x}, \underline{y}).$$

Mixed Finally, the mixed policies combine the retrospective evaluation of the history-based method and the prospective evaluation of the model-based method. In particular, we use the **model**-based approach for non-train $\hat{\pi}$ (*i.e.*, scoring with $S^{\text{EXP}(y)}$) and the **history**-based approach for **train** policy π_{train} (*i.e.*, scoring with S^{EMP}):

$$\begin{aligned} S(\hat{\pi}; \underline{x}, \underline{y}) &= S^{\text{EXP}(y)}(\hat{\pi}; \underline{x}, \underline{y}), \\ S(\pi_{\text{train}}; \underline{x}, \underline{y}) &= S^{\text{EMP}}(\pi_{\text{train}}; \underline{x}, \underline{y}), \\ \pi^* &= \arg \max_{\pi \in [\hat{\pi}, \pi_{\text{train}}]} S(\pi; \underline{x}, \underline{y}). \end{aligned}$$

For $t = 0$, we always select π_{train} to ensure we have an empirical mean for π_{train} . Table 1 provides an overview of the query policies described in the preceding sections.

4 A Grammatical Model for Phonotactics

We implement and test our approach for a simple categorical model of phonotactics. The grammar consists of two components. First, a finite set of phonological feature functions $\{\phi_i\} : \Sigma^+ \mapsto \{0, 1\}$; if $\phi_i(x) = 1$ we say that feature i is **active** for string x . This set is taken to be universal and available to the learner before any data are observed. Second, a set of binary values $\theta = \{\theta_i\}$, one for each feature function; if $\theta_i = 1$ then feature i is **penalized**. In our simple categorical model, a string is grammatical if and only if no feature active for it is penalized. θ thus determines the language: $L = \{x : \sum_i \theta_i(x)\phi_i(x) = 0\}$. We assume a factorizable prior distribution over which features are active: $p(\theta) = \prod_{\theta_j \in \theta} p(\theta_j)$. To enable mathematical tractability, we also incorporate a noise term α which causes the learner to perceive a judgment from the informant as noisy (reversed) with probability $1 - \alpha$.

This model is based on a decades-long research tradition in theoretical and experimental phonology into what determines the range and frequency of possible word forms in a language. A consensus view of the topic is that speakers have fine-grained judgments about the acceptability of nonwords (for example, most speakers judge *blick* to be more acceptable than *bnick*; Chomsky and Halle, 1968), and that this knowledge can be decomposed into the independent, additive effects of multiple prohibitions on specific sequences of sounds (in phonological theory, termed MARKEDNESS constraints). Further, speakers form these generalizations at the level of the phonological feature, since they exhibit structured judgments that distinguish between different unattested forms: speakers systematically rate *bnick* as less English-like than *bzick*, despite no attested words having initial *bn-* or *bz-*. We reflect this knowledge in our generative model: to determine the distribution of licit strings in a language, we first sample some parameters which govern subsequences of features which are penalized in the language.

In our model we take $\{\phi_i\}$ to be a collection of phonological **feature trigrams**: an ordered triple of three phonological features with values that pick out some class of trigrams of segments in the language (see §5.1 for more details and examples). Since our phonotactics are variants on vowel harmony, these featural trigrams are henceforth assumed to be relativized to the vowel tier, regulat-

Basic Policy	Quantity Maximized	Hybrid Policy	Basic Choices Π	Method	Metric V	Basic Policy Selection	
						π_{train} score	Non-train score
π_{train}	–	$\pi_{\text{eig-hist}}$		History		S^{EMP}	S^{EMP}
π_{unif}	–	$\pi_{\text{eig-model}}$	$\pi_{\text{train}}, \pi_{\text{eig}}$	Model	Info gain (VIG, Eq 1)	$S^{\text{EXP}(x)}$	$S^{\text{EXP}(y)}$
$\pi_{\text{label-ent}}$	Label entropy			Mixed		S^{EMP}	$S^{\text{EXP}(y)}$
π_{eig}	Expected info gain	$\pi_{\text{eig-mixed}}$					

(a) Basic policies (§3.1).

(b) Hybrid policies (§3.2).

Table 1: Summary of query policies (§3). S^{EMP} refers to empirical mean. $S^{\text{EXP}(y)}$ and $S^{\text{EXP}(x)}$ refer to the expectation metrics for the non-train $\hat{\pi}$ and train π_{train} strategies, respectively. Basic policies select inputs to query the informant. Hybrid policies choose between a set of basic policies Π by scoring them with a metric V and one of the scoring functions.

ing vowel qualities in three adjacent syllables. In order to capture generalizations that may hold differently in edge-adjacent vs. word-medial position, we pad the representation of each word treated by the model with a boundary symbol “#” — omitted generally in this paper, for simplicity — which bears the [+ word boundary] feature that the tri-gram constraints can refer to (following the practice of Hayes and Wilson, 2008, inspired by Chomsky and Halle, 1968).

4.1 Implementation details

Our general approach and specific model create several computational tractability issues that we address here. First, all policies aside from π_{train} and π_{unif} in principle require search for an optimal string x within \mathcal{X} . In practice, we consider $\mathcal{X} = \Sigma^+ \{2, 5\}$, *i.e.*, \mathcal{X} is the set of strings with 2-5 syllables. This resulting set is still very large, so we approximate the search over \mathcal{X} by uniformly sampling a set of k candidates and choosing the best according to V . We sample candidates by uniformly sampling a length, then uniformly sampling each syllable from the inventory of possible onset-vowel combinations in the language (with replacement). We then de-duplicate candidates and filter \underline{x} , excluding previously observed sequences and those that were accidental duplicates with items in the test set.

Second, although the model parameters θ are independent in the prior, conditioning on data renders them conditionally dependent and computing with the true posterior is in general intractable. To deal with this, we use mean-field variational Bayes to approximate the posterior as $p(\theta \mid \underline{x}, \underline{y}) \approx \prod_{\theta_j \in \theta} q(\theta_j = 1 \mid \underline{x}, \underline{y})$. We use this approximation to both estimate the model’s posterior (used by $\pi_{\text{label-ent}}$ and π_{eig}) and to make predictions about individual new examples. See Appendix D for details.

5 Experiments

We now describe our experiments for evaluating the different query policies. We evaluate on two types of languages. We call the first the ATR Vowel Harmony language (§5.1), which has grammar that regulates the distribution of types of vowels, inspired by those found in many languages of the world. The purpose of evaluating on this language is to evaluate how well our new approach, and specifically the various non-baseline query policies, work on naturalistic data. We also evaluate on a set of procedurally-generated languages (§5.2) that are matched on statistics to ATR Vowel Harmony, *i.e.*, they have the same number of feature tri-grams that are penalized, but differ in *which*. This second set of evaluations aims to determine how robust our model is to typologically-unusual languages, so we can be confident that any success in learning ATR Vowel Harmony is attributable to our procedure, rather than a coincidence of the typologically-natural vowel harmony pattern.

These experiments lead to three sets of analyses: in the first (§5.4), we both select hyperparameters and evaluate on procedurally-generated languages through k -fold cross validation. These results can be interpreted as an in-distribution analysis of the query policies. In the second set of results (§5.5), we evaluate the policies out-of-distribution by selecting hyperparameters on the procedurally-generated languages and evaluating on the ATR Vowel Harmony language. In the last analysis (§5.6), we evaluate the upper bound of policy performance by selecting hyperparameters and evaluating on the same language, ATR Vowel Harmony.

5.1 ATR Vowel Harmony

We created a model language whose words are governed by a small set of known phonological principles. Loosely inspired by harmony systems

common among Niger-Congo and Nilo-Saharan languages spoken throughout Africa, the vowels in this language can be divided into two classes, defined with respect to the phonological feature Advanced Tongue Root (ATR); for typological data, see Casali (2003, 2008, 2016); Rose (2018), among others. In this language, vowels that are [+ATR] are {i, e}, and have pronunciations that are more peripheral in the vowel space; those that are [-ATR] are {ɪ, ɛ}, and are more phonetically centralized. For the sake of simplicity, we restrict the simulated language to only have front vowels. A fifth vowel in the system, [a], is not specified for ATR. This language has consonants {p, t, k, q}, which are distributed freely with respect to one another and to vowels with the exception that syllables must begin with exactly consonant and must contain exactly one vowel, a typologically common restriction. Since consonants are not regulated by the grammar we are working with, the three binary features (leaving out [word boundary]) create a set of 512 possible feature trigrams which characterize the space of all possible strings in the language. The syllable counts of words follows a Poisson distribution with $\lambda = 2$.

The single rule active in this language governs the distribution of vowels specified for the feature [ATR]: vowels in adjacent syllables had to have the same [ATR] specification. This means that vowel sequences in a word can be [i...e] or [ɛ...ɪ], but not [e...ɛ] or [e...ɪ]. Since [a] is not specified for ATR, it creates boundaries that allow different ATR values to exist on either side of it: for example, while the vowel sequence [e...ɛ] is not permitted, the sequence [e...a...ɛ] is allowed, because the ATR-distinct vowels are separated by the unspecified [a]. This yielded sample licit words like [katipe], [tɛpi], and [qekati], and illicit ones [kɛkiqa], [titaqikɛ], and [qiqika].

Feature trigrams were composed of triples of the features and specifications shown in Appendix Table 3, any one of which picks out a certain set of vowel trigrams in adjacent syllables.

Data We sampled 157 unique words as the lexicon L , and a set of 1,010 random words, roughly balanced for length, as a test set. The model was provided with the set of features in Appendix Table 3, and restrictions on syllable structure for use in the proposal distribution.

Informant The informant was configured to reject any word that contained vowels in adjacent syllables that differed in ATR specification (like [pekite] or [qetatikipe]), and accept all others.

5.2 Procedurally-Generated Languages

We also experimented with languages that share the same feature space, and thus the same set of 512 feature trigrams, as ATR Vowel Harmony (§5.1) but were *procedurally generated* by sampling 16 of the 512 total feature trigrams to be “on” (*i.e.*, penalized) and set all others to be off, creating languages with different restrictions on licit vowel sequences in adjacent syllables.

Data For each “language” (*i.e.*, set of sampled feature trigrams to be penalized), we carried out a procedure to sample the lexicon L , as well as evaluation datasets. For each set of 16 θ values representing penalized phonological feature trigrams, we created random strings as in Experiment 1, filtering them to ensure that the train and test set are of equal size, and the test set is balanced for length of word and composed of half acceptable and half unacceptable forms.

5.3 Experimental Set-Up

Hyperparameters The model has several free parameters: a noise parameter α that represents the probability that an observed label is correct (versus noisy), and θ_{prior} , the prior probability of a feature being *on* (penalized), *i.e.*, $p_{\text{prior}}(\theta_j = 1)$. There are also hyperparameters governing the optimization of the model: we denote by s the number of optimization steps in the variational update.¹ When $s = \infty$, we optimize until the magnitude of the change in θ is less than or equal to an error threshold $\epsilon = 2e^{-7}$. We also experiment with $s = 1$, in which we perform a single update.

We ran a grid-search over the parameter space of $\log(\log(\alpha)) \in \{0.1, 0.25, 0.5, 1, 2, 4, 8\}$, $\theta_{\text{prior}} \in \{0.001, 0.025, 0.05, 0.1, 0.2, 0.35\}$, and $s \in \{1, \infty\}$. We ran 10 random seeds (9 for the procedurally generated languages)² and all query policies in Table 1 for each hyperparameter setting. Each experiment was run for 150 steps.

For non-train policies, we generated $k = 100$ candidates from \mathcal{X} .

¹These optimization parameters govern both the model’s learning and the evaluation of candidate queries for prospective strategies, *i.e.*, π_{eig} , and the hybrid strategies.

²For the generated languages, seed also governed the “language,” *i.e.*, phonological feature trigrams sampled as “on.”

Evaluation At each step, we compute the AUC (area under the ROC curve) on the test set. We then compute the mean AUC value across steps, which we refer to as the **mean-AUC**; a higher mean-AUC indicates more efficient learning. We report the median of the mean-AUC values over seeds.

5.4 In-Distribution Results

Assessing the in-distribution results, shown in the left column of Figure 2, we see that interactive elicitation is on par with, if not higher than, baseline strategies (top left plot). The difference between the *train* and *uniform* baselines was not significant according to a two-sided paired *t*-test, and the only strategy that performed significantly better than *train* after correcting for multiple comparisons was *Info. gain / train (model)*. This difference is more visually striking in the plot of average AUC over time (middle left plot), where *Info. gain / train (model)* both ascends faster, and asymptotes earlier, than *train*, although with greater variance across runs. In the bottom left plot of Figure 2, we see that the numerically-best-performing *Info. gain / train (model)* strategy moves rather smoothly from an initial *train* preference to an *Info. gain* preference as learning progresses. That is, information in known-good words is initially helpful, but quickly becomes less useful as the model learns more of the language and can generate more targeted queries.

5.5 Out-Of-Distribution Results

The out-of-distribution analysis on the ATR Vowel Harmony language found greater variance of median mean-AUC between strategies, and also greater variance within strategies across seeds (top center plot). We note that this performance is lower than what is found in the upper-bound analysis, since the hyperparameters (listed in Appendix Table 2) were chosen based on the pooled results of the procedurally-generated languages. As in the in-distribution analysis, we found no statistical difference between the two baselines, nor between the *Info. gain* strategy and *uniform*, although *Info. gain* performed numerically better. In terms of average AUC over time (middle center plot), we find again that the top two non-baseline strategies rise faster and peak earlier than *uniform*, but exhibit greater variance.

5.6 Upper Bound Results

Greedily selecting for the best test performance in a hyperparameter search conducted on ATR Vowel

Harmony yields superior performance compared to the out-of-distribution analysis hyperparameters, as seen in the top right plot in Figure 2. Appendix Table 2 lists the hyperparameter values used. However, we found no significant difference between the stronger baseline (*uniform*) and any other strategy after correcting for multiple comparisons.

6 Related Work

The goal of **active learning** is to improve learning efficiency by allowing models to choose which data to query an oracle about (Zhang et al., 2022). *Uncertainty sampling* (Lewis and Gale, 1994) methods select queries for which model uncertainty is highest. Most closely related are uncertainty sampling methods for probabilistic models, including least-confidence (Culotta and McCallum, 2005), margin sampling (Scheffer et al., 2001), and entropy-based methods.

Disagreement-based strategies query instances that maximize disagreement among a group of models (Seung et al., 1992). The distribution over a single model’s parameters can also be treated as this “group” of distinct models, as has been done for neural models (Gal et al., 2017). Such methods are closely related to the feature entropy querying policy that we explore.

Another class of *forward-looking methods* incorporates information about how models would change if a given data-point were observed. Previous work includes methods that sample instances based on expected loss reduction (Roy and McCallum, 2001), expected information gain (MacKay, 1992), and expected gradient length (Settles et al., 2007). These methods are closely related to the policies based on information-gain that we explore.

Our hybrid policies are also related to previous work on dynamic selection between multiple active learning policies, such as DUAL (Donmez et al., 2007), which dynamically switches between density and uncertainty-based strategies.

The model we propose is also related to a body of work in computational and theoretical linguistics focused on **phonotactic learning**. Much of this work, largely inspired by Hayes and Wilson (2008), seeks to discover and/or parameterize models of phonotactic acceptability on the basis of only positive data, in line with common assumptions about infant language acquisition (Albright, 2009; Adriaans and Kager, 2010; Linzen and O’Donnell, 2015; Futrell et al., 2017; Mirea and Bicknell, 2019;

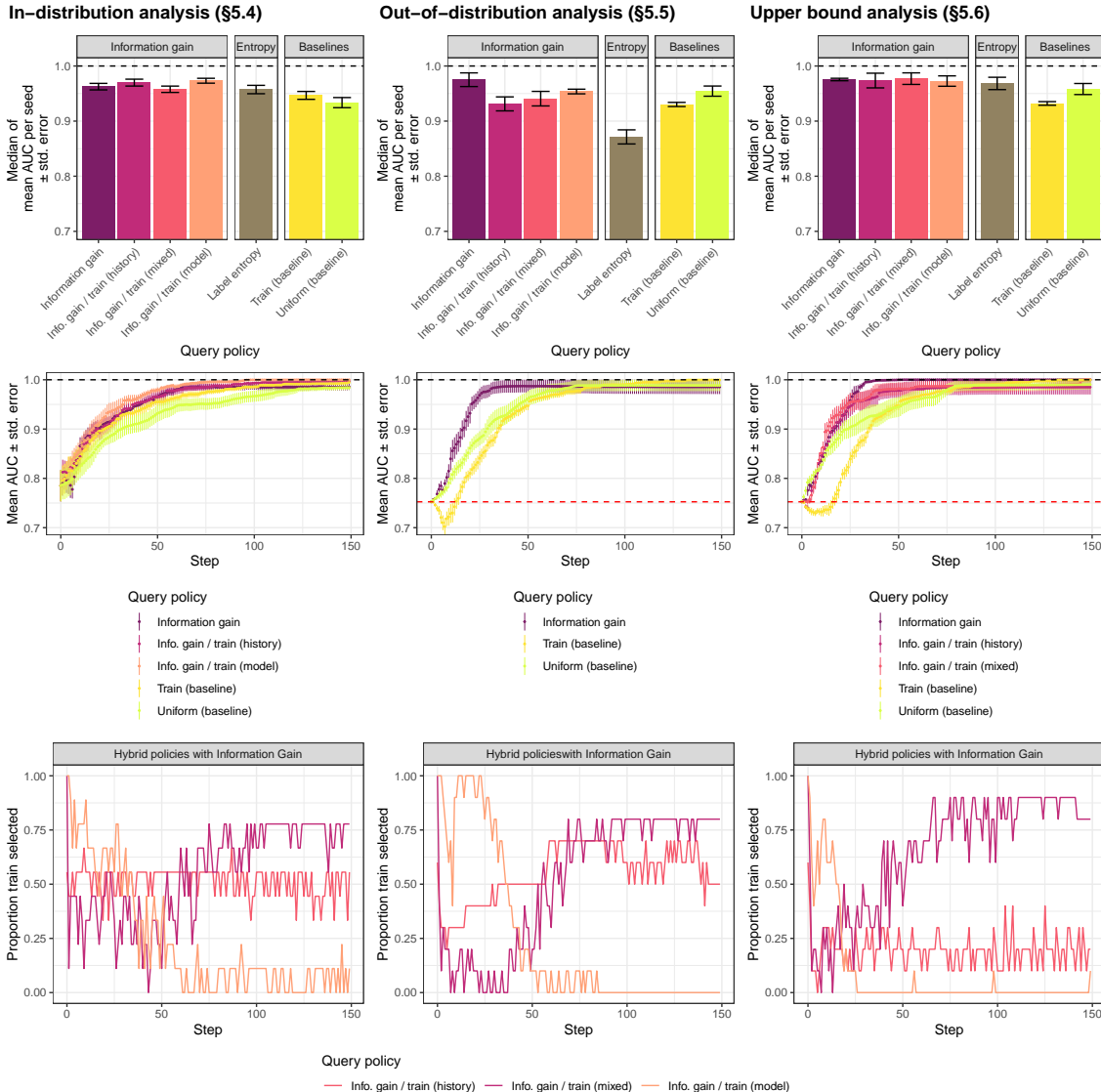


Figure 2: We report three analyses of the toy ATR Vowel Harmony language and our procedurally-generated languages: in-distribution (left column, see §5.4), out-of-distribution (center column, see §5.5), and an upper-bound assessment (right column, see §5.6). For each, we report the median and standard error of the mean-AUC over steps aggregated across runs (top row; numerical values and hyperparameters reported in Appendix Table 2), average AUC at each step aggregated across runs (middle row), and at each step the proportion of runs where the basic *train* strategy was selected by the hybrid strategies (bottom row). **Results:** In terms of median mean-AUC (top row), our query strategies are numerically on par with, if not beating, the stronger of the two baseline conditions; statistically, only the difference between *Info. gain / train (model)* and *uniform* was significant in the in-distribution analysis (top left). Average AUC over time (middle row) shows a similar pattern across all three analyses, with the non-baseline strategies rising faster and asymptoting sooner than baseline strategies, but usually with greater variance. Finally, though all hybrid strategies prefer non-*train* some portion of the time, the *Info. gain / train (model)* exhibits an interpretable shift from early preference for *train* data to later preference for its own synthesized queries in all three analyses.

Gouskova and Gallagher, 2020; Mayer and Nelson, 2020; Dai et al., 2023; Dai, to appear). Our work differs from these in that we are explicitly not seeking to model phonotactic learning from the infants’

point of view, instead drawing inspiration from the strategy of a linguist working with a competent native speaker to discover linguistic structure via iterated querying. Practically, this means that our

model can make use of both positive and negative data, and also takes an active role in seeking out the data it will learn from.

7 Conclusion

We have described a method for parameterizing a formal model of language via efficient, iterative querying of a black box agent. We demonstrated that on an in-distribution set of toy languages, our query policies consistently outperform baselines numerically, including a statistically-significant improvement for the most effective policy. The model struggles more on out-of-distribution languages, though in all cases the query policies are numerically comparable to the best baseline. We note that a contributing factor to the difficulty of the query policies consistently achieving a *significantly* higher performance than baselines is the small number of seeds, which exhibit nontrivial variance, particularly in hybrid policies. Future work may address this with more robust experiments.

Acknowledgements

Thanks to members of the audience at Interactions between Formal and Computational Linguistics (IFLG) Seminar hosted by the *Linguistique Informatique, Formelle et de Terrain* group, as well as two anonymous SCiL reviewers, for helpful questions and comments.

We acknowledge the following funding sources: MIT-IBM Watson AI Lab (CB, RL), NSF GRFP grant number 2023357727 (AR), a MIT Shillman Fellowship (AR), a MIT Dean of Science Fellowship (AM), and NSF IIS-2212310 (JA, AR).

References

- Frans Adriaans and René Kager. 2010. Adding generalization to statistical learning: The induction of phonotactics from continuous speech. *Journal of Memory and Language*, 62(3):311–331.
- Adam Albright. 2009. Feature-based generalisation as a source of gradient acceptability. *Phonology*, 26(1):9–41.
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.
- Roderic F Casali. 2003. [atr] value asymmetries and underlying vowel inventory structure in niger-congo and nilo-saharan.
- Roderic F Casali. 2008. Atr harmony in african languages. *Language and linguistics compass*, 2(3):496–549.
- Roderic F Casali. 2016. Some inventory-related asymmetries in the patterning of tongue root harmony systems. *Studies in African Linguistics*, pages 96–99.
- Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper & Row New York.
- Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2, AAAI’05*, page 746–751. AAAI Press.
- Huteng Dai. to appear. An exception-filtering approach to phonotactic learning. *Phonology*.
- Huteng Dai, Connor Mayer, and Richard Futrell. 2023. Rethinking representations: A log-bilinear model of phonotactics. *Proceedings of the Society for Computation in Linguistics*, 6(1):259–268.
- Pinar Donmez, Jaime G. Carbonell, and Paul N. Bennett. 2007. [Dual strategy active learning](#). In *Proceedings of the 18th European Conference on Machine Learning, ECML ’07*, page 116–127, Berlin, Heidelberg. Springer-Verlag.
- 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.
- Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, page 1183–1192. JMLR.org.
- Maria Gouskova and Gillian Gallagher. 2020. Inducing nonlocal constraints from baseline phonotactics. *Natural Language & Linguistic Theory*, 38:77–116.
- Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.
- John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.
- David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR ’94*, pages 3–12, London. Springer London.
- Tal Linzen and Timothy O’Donnell. 2015. A model of rapid phonotactic generalization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1126–1131.

David J. C. MacKay. 1992. [Information-Based Objective Functions for Active Data Selection](#). *Neural Computation*, 4(4):590–604.

Connor Mayer and Max Nelson. 2020. Phonotactic learning with neural language models. *Society for Computation in Linguistics*, 3(1).

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.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al. 2022. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*.

Sharon Rose. 2018. Atr vowel harmony: new patterns and diagnostics. In *Proceedings of the Annual Meetings on Phonology*, volume 5.

Nicholas Roy and Andrew McCallum. 2001. [Toward optimal active learning through monte carlo estimation of error reduction](#). In *International Conference on Machine Learning*.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *Advances in Intelligent Data Analysis*, pages 309–318, Berlin, Heidelberg. Springer Berlin Heidelberg.

Burr Settles, Mark Craven, and Soumya Ray. 2007. [Multiple-instance active learning](#). In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

H. S. Seung, M. Opper, and H. Sompolinsky. 1992. [Query by committee](#). In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, page 287–294, New York, NY, USA. Association for Computing Machinery.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. [A survey of active learning for natural language processing](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Phonological features for Toy Languages

As described in §5.1, the ATR Vowel Harmony language is based on the categorization of vowels as [+ATR], [-ATR], or unspecified. The features [high] and [low] also serve to distinguish vowels in the language, but are not governed by a phonotactic. In contrast, any of the 512 logically possible

trigrams of specified phonological features may be penalized for the procedurally-generated languages. Table 3 displays the phonological features for each of the vowels in the languages.

B Hyperparameters for out-of-distribution and upper-bound analyses

In §5.3, we described the hyperparameters of our grammatical model and the process by which values were selected for the out-of-distribution analysis. These selected hyperparameter values are presented in Table 2.

C Query Policy Implementation

We now revisit the query strategies introduced in §3 and describe how they are implemented for the model described in §4. In particular, under the described generative model, $p(y = 1 | x, \underline{x}, \underline{y}) = \prod_{j \in \phi(x)} q(\theta_j = 0 | x, \underline{x}, \underline{y})$, as described above.

Let $q_y = \prod_{j \in \phi(x)} q(\theta_j = 0 | x, \underline{x}, \underline{y})$, *i.e.*, q_y is the probability of label $y = 1$ for input x under the variational posterior; this is equivalent to the probability of all features in $\phi(x)$ being “off”. Let $q_{\theta_j} = q(\theta_j = 1 | \underline{x}, \underline{y})$ indicate the probability of parameter θ_j being “on” (*i.e.*, penalized) under the current variational $q(\theta)$. For this model, the quantities used by the query policies in §3 are computed as follows:

Label Entropy Policy $\pi_{\text{label-ent}}$ selects x^* according to:

$$x^* = \arg \max_{x \in \mathcal{X}} H(q_y), \text{ where}$$

$$H(p(y | x, \underline{x}, \underline{y})) = -q_y \log q_y - (1 - q_y) \log(1 - q_y).$$

Expected Information Gain Policy π_{eig} selects x^* according to:

$$x^* = \arg \max_{x \in \mathcal{X}} V_{\text{IG}}(x, y = 1; \underline{x}, \underline{y}) \cdot q_y + V_{\text{IG}}(x, y = 0; \underline{x}, \underline{y}) \cdot (1 - q_y),$$

where V_{IG} is given by

$$V_{\text{IG}}(x, y; \underline{x}, \underline{y}) = \sum_{j \in |\theta|} \left(H(q(\theta_j | \underline{x}, \underline{y})) - H(q(\theta_j | x, y, \underline{x}, \underline{y})) \right),$$

Out-of-distribution analysis						Upper-bound analysis					
Policy	$\log(\log(\alpha))$	prior	s	Median mean-AUC	Std. err.	Policy	$\log(\log(\alpha))$	prior	s	Median mean-AUC	Std. err.
Info. gain / train (model)	0.5	0.1	∞	0.973	0.004	Info. gain / train (mixed)	0.25	0.1	∞	0.977	0.010
Info. gain / train (history)	1	0.1	∞	0.970	0.006	Information gain	0.1	0.025	∞	0.975	0.002
Info. gain / train (mixed)	2	0.2	∞	0.969	0.005	Info. gain / train (history)	0.1	0.05	∞	0.974	0.013
Information gain	0.25	0.025	∞	0.966	0.004	Info. gain / train (model)	1	0.001	1	0.973	0.009
Label entropy	0.1	0.1	∞	0.964	0.009	Label entropy	0.5	0.05	1	0.968	0.011
Train (baseline)	1	0.1	∞	0.947	0.007	Uniform (baseline)	0.5	0.025	1	0.958	0.010
Uniform (baseline)	1	0.1	1	0.940	0.008	Train (baseline)	8	0.35	1	0.932	0.003

Table 2: Hyperparameters for the out-of-distribution analysis (§5.5) and upper-bound analysis (§5.6).

and H is given by

$$H(q(\theta_j)) = -q\theta_j \log q\theta_j - (1 - q\theta_j) \log(1 - q\theta_j).$$

D Derivation of the Update Rule

We want to compute the posterior $p(\theta|y, x, \alpha)$, which is intractable. Thus, we approximate it with a variational posterior, composed of binomial distributions for each θ_i . We further assume that the individual dimensions of the posterior (the individual components of θ) have values that are not correlated. This allows us to perform coordinate ascent on each dimension of the posterior separately; thus we express the following derivation in terms of $q(\theta_i)$, where i is the index in the feature n -gram vector.

The variational posterior is optimized to minimize the KL divergence between the true posterior $p(\theta|X, Y, \alpha)$ and $q(\theta)$; we do this by maximizing the ELBO.

The coordinate ascent update rule for each dimension of the posterior, that is, for each latent variable, is:

$$q(\theta_i) \propto \exp [\mathbb{E}_{q^{-i}} \log p(\theta_i, \theta_{-i}, y, x)].$$

Given the generative process, we can rewrite:

$$p(\theta_i, \theta_{-i}, y, x) = p(\theta_i) \cdot p(\theta_{-i}) \cdot p(y|x, \theta_i, \theta_{-i}).$$

	[high]	[low]	[ATR]
i	+	-	+
ɪ	+	-	-
e	-	-	+
ɛ	-	-	-
a	-	+	0

Table 3: Phonological features for vowels used in the toy languages. The feature [word boundary] is omitted for simplicity, as it has the value ‘-’ for all segments.

$\mathbb{E}_{q^{-i}} \log p(\theta_{-i})$ is assumed to be constant across values of θ_i (expressing the lack of dependence between parameters), so we can rewrite the update rule as:

$$q(\theta_i) \propto \exp [\mathbb{E}_{q^{-i}} [\log p(\theta_i) + \log p(y|x, \theta_i, \theta_{-i})]].$$

Further, since $\log p(\theta_i)$ is constant across values of q^{-i} , we can rewrite it once more:

$$q(\theta_i) \propto \exp [\log p(\theta_i) + \mathbb{E}_{q^{-i}} \log p(y|x, \theta_i, \theta_{-i})].$$

Since our approximating distribution is binomial, we describe in turn the treatment of each of the two possible values of θ . First, we derive the update rule for when the label y is acceptable ($y = 1$).

We know that there are two subsets of q^{-i} cases where this can happen. In α proportion of them, y is a correct label, which can only happen when $\theta_j = 0$ for all $j \neq i \in \phi(x)$. This occurs with probability $p_{\text{all_off}} = \prod_{j \neq i \in \phi(x)} q(\theta_j = 0)$. There is also, then, the $1 - \alpha$ proportion of cases in which y is an incorrect label, and the true judgement is unacceptable. Under this assumption, at least 1 feature is on, which occurs with probability $1 - p_{\text{all_off}}$.

We can rewrite the expectation term to get approximate probabilities for both the $\theta_i = 0$ and $\theta_i = 1$ cases when $y = 1$:

$$q(\theta_i = 0) \propto \exp [\log p(\theta_i = 0) + (p_{\text{all_off}} \cdot \log \alpha + (1 - p_{\text{all_off}}) \cdot \log(1 - \alpha))].$$

If $\theta_i = 1$, we know that $\log p(y|x, \theta_i, \theta_{-i}) = \log(1 - \alpha)$ for all q^{-i} , since we know that y must be a noisy label. Thus:

$$q(\theta_i = 1) \propto \exp [\log p(\theta_i = 1) + \log(1 - \alpha)].$$

We can normalize these quantities to get a proper probability distribution, i.e. we can set $q(\theta_i = 1)$ to the following quantity:

$$q(\theta_i = 1) := \frac{q(\theta_i = 1)}{q(\theta_i = 1) + q(\theta_i = 0)}.$$

Using the expression $q(\theta_i)$ as shorthand for $q(\theta_i = 1)$, this results in the following update rule:

$$q(\theta_i = 1) = \sigma \left(\log p(\theta_i) - \log(1 - p(\theta_i)) - p_{\text{all_off}} \cdot \log \frac{\alpha}{1 - \alpha} \right).$$

In practice, we update over batches of inputs/outputs rather than single datapoints, *i.e.*,

$$\mathbf{m}_{\mathbf{i}, \mathbf{j}} = \sum_{j' \neq j \in \phi(x_i)} \log(1 - p(\theta_{j'})) + \log \log \frac{\alpha}{1 - \alpha},$$

$$q(\theta_j) = \sigma(\log p(\theta_j) - \log(1 - p(\theta_j)) - \sum_{i < t} y_i \cdot \exp(\mathbf{m}_{\mathbf{i}, \mathbf{j}})).$$

We update each $q(\theta_j)$ either for a fixed number of steps s , or until convergence, *i.e.*, when:

$$\left| \sum_{j \in \theta} q_j^{\delta+1} - q_j^{\delta} \right| < \epsilon,$$

where ϵ is an error threshold.

Morphologically simplex D-quantifiers are strictly 2-local

Thomas Graf

Department of Linguistics & Institute for Advanced Computational Science
Stony Brook University
mail@thomasgraf.net

Abstract

Even though languages can express a wide range of quantifiers, only a small number are ever realized as morphologically simplex determiners: *every*, *no*, *some*, and *most*. This is puzzling because I) *most* is much more complex than the other three, and II) quantifiers like *an even number* are simpler than *most* yet cannot belong to this class. Building on concepts from subregular complexity, I present a new way of measuring a quantifier's complexity in terms of its verification pattern. The quantifiers *every*, *no*, *some*, and *most* all have strictly 2-local (SL-2) verification patterns, but quantifiers like *an even number* do not. This suggests that subregular complexity, and in particular strict locality, plays a crucial role for how much meaning can be packed into morphologically simplex expressions.

1 Introduction

The literature on generalized quantifiers (see Keenan and Westerstahl 1996, Peters and Westerstahl 2006 and references therein) considers a wide range of quantificational expressions, from *every*, *no*, and *some* to *not all*, *all but one*, *most*, *at most half*, *an even number*, *a third*, *between two and eight*, or *more - than*. It is noteworthy, though, that across languages the majority of these expressions are structurally complex, involving multiple words or morphemes. For instance, there seems to be no language with a single word that has the same meaning as *not all*. This is particularly well-documented in the case of determiners. Among *D-quantifiers*, i.e. quantifiers that function as determiners, the only simplex ones (*modulo* agreement markers) are realizations of *every*, *no*, *some*, and *most*, although not all of them are instantiated in every language.

Surprising as this may be, it becomes even more puzzling once one considers the complexity of these quantifiers. Semantic automata theory

(van Benthem, 1986; Steinert-Threlkeld and Icard, 2013) allows us to determine a quantifier's position in the Chomsky-hierarchy of string languages (Chomsky, 1956, 1959; Chomsky and Schützenberger, 1963). Many quantifiers are regular, including simplex *every*, *no*, and *some*, but also the morphologically complex expressions *not all*, *all but one*, and *an even number*. On the other hand, *most* belongs to the more complex class of context-free string languages. If *most* can be a morphologically simplex D-quantifier (MSDQ), why isn't this possible for some quantifiers of lower complexity?

Recently, I set out to refine this picture in Graf (2019b) by drawing from work on the subregular complexity of patterns in phonology, morphology, and syntax (see Chandler 2017, 2022, Heinz 2018, Dolatian and Heinz 2020, Graf 2022a,b, Hanson 2023a,b, and references therein). I argued that among the regular quantifiers, *every*, *no*, *some*, *not all*, and *all but one* are particularly simple because they belong to the subregular class of tier-based strictly local languages (Heinz et al., 2011; Lambert and Rogers, 2020), whereas *an even number* does not. While this explains how *an even number* differs from these quantifiers, it still does not explain why *not all* and *all but one* cannot be MSDQs, and it actually widens the complexity gap between *most* and the other MSDQs.

In this paper, I propose that the contradictory complexity results are resolved by adopting *verification patterns* as a new string model of quantifier interpretation. A verification pattern for quantifier Q encodes instructions for how the elements of the domain can be arranged to easily determine whether the statement expressed by Q is true. The complexity of Q is equated with the complexity of the simplest possible verification pattern for Q . The MSDQs *every*, *no*, *some*, and *most* all have strictly 2-local verification patterns, but *an even number* does not. Verification patterns thus place the attested MSDQs within the same complexity

class while correctly excluding many other quantifiers.

The paper is laid out as follows. After a brief discussion of the notion of *quantifier languages* from semantic automata theory (Sec. 2.1), I define what it means for a string language to be strictly k -local (Sec. 2.2). I then define verification patterns as an alternative to quantifier languages (Sec. 3.1) and show that all MSDQs have SL-2 verification patterns, in particular *most* (Sec. 3.2 and 3.3). But not all quantifiers with SL-2 verification patterns are MSDQs, and Sec. 3.4 proposes several conditions that separate the MSDQs from the other quantifiers in this class. The paper concludes with some tentative observations on how this approach could be extended to handle infinite domains, various pragmatic effects, capture typological frequency effects, and cognitive parallels to syntax (Sec. 4).

2 Background

2.1 Semantic automata and the typology of quantifiers

Generalized quantifiers like *every*, *no*, *some*, and *most* are formally modeled as type $\langle 1, 1 \rangle$ quantifiers, i.e. as functions that take two sets A and B as arguments and return a truth value depending on whether a specific relation holds of A and B .

Example 1. The quantifier *every* corresponds to the function $f_{\text{every}} : E \times E \rightarrow \{0, 1\}$ (where E is some fixed set of entities) such that $f_{\text{every}}(A, B) = 1$ iff $A \subseteq B$. In the sentence *every cat sneezed*, A is the set of cats and B is the set of entities that sneezed. The sentence is true iff the set of cats is a subset of the set of sneezers. \lrcorner

The *semantic automata* approach (van Benthem, 1986; Steinert-Threlkeld and Icard, 2013) makes it possible to recast any type $\langle 1, 1 \rangle$ quantifier Q as a string language L_Q over the alphabet $\{0, 1\}$. We also call L_Q a *quantifier language*. Intuitively, one constructs a binary string s_B^A such that each position i of s_B^A corresponds to a distinct element $a_i \in A$, and the symbol at position i is 1 if $a_i \in B$ and 0 otherwise. Given a quantifier Q , then, $f_Q(A, B) = 1$ iff $s_B^A \in L_Q$. Crucially, for all quantifiers discussed in this paper this must hold no matter how elements are associated to positions, so $s_B^A \in L_Q$ iff L_Q contains every permutation of s_B^A .

Example 2. Continuing the previous example, suppose that the discourse salient set A of cats consists

of Mocha, Murli, and Cinderella, whereas the set B of sneezers consists of Mocha and Mary. In this scenario, it is false that every cat sneezed, and the semantic automata approach expresses this as follows.

First, Mocha is replaced with 1, whereas Murli and Cinderella are each mapped to 0. With one 1 and two 0s, we can build three binary strings: 100, 010, and 001. The quantifier language of *every* consists of all strings that do not contain 0. For if some element $a \in A$ is replaced by 0, then $a \notin B$ and thus $A \not\subseteq B$. And in the other direction, $A \subseteq B$ entails that there is at least one a such that $a \notin B$, and hence s_B^A must contain at least one 0. None of the three binary strings above are members of L_{every} , and thus *every cat sneezed* is correctly predicted to be false for the specific scenario described above. \lrcorner

With semantic automata, the cognitive complexity of quantifiers can be measured in terms of the complexity of the computational machinery that is needed to generate the corresponding quantifier languages. Tab. 1 lists some well-known complexity results. Note that many quantifier languages actually belong to a proper subclass of the class listed in the table. For example, *most* could be more adequately classified as a deterministic context-free language, or even more tightly as a one-counter language. In Graf (2019b) I showed that *every* and *no* are *strictly 1-local* (SL-1), whereas *some*, *not all*, and *exactly one* are *tier-based strictly 2-local* (TSL-2) — a large reduction in complexity with connections to phonology (McMullin and Hansson, 2015; McMullin, 2016; Jardine and Heinz, 2016; Burness et al., 2021; Mayer, 2021). These refinements do not change the fact, though, that complexity tells us little about what quantifiers may be MSDQs.

Only four MSDQs are attested across languages: *every*, *no*, *some*, and *most* (English *one* does not belong in this category because it is a numeral, and Russian has morphologically simplex *half* but its syntactic behavior is that of a noun rather than a determiner). Why should these be the only MSDQs? Why is it impossible for, say, *an even number of* to ever be realized as an MSDQ? Complexity considerations make this even more puzzling: the class of attested MSDQs contains two that are SL-1, one that is TSL-2, and one that isn't even regular, while excluding many quantifiers of similar or lesser complexity.

Quantifier	Definition	String constraint	Complexity	MSDQ?
every	$A \subseteq B$	no 0	regular	Yes/No
every (existential import)	$A \subseteq B \& A \neq \emptyset$	no 0 and at least one 1	regular	Yes/No
no	$A \cap B = \emptyset$	no 1	regular	Yes
not all	$A \not\subseteq B$	at least one 0	regular	No
some/at least one	$A \cap B \neq \emptyset$	at least one 1	regular	Yes
exactly one	$ A \cap B = 1$	exactly one 1	regular	No
an even number of	$ A \cap B \bmod 2 = 0$	an even number of 1s	regular	No
half	$ A \cap B = A - B $	an equal number of 1s and 0s	context-free	No
most	$ A \cap B > A - B $	more 1s than 0s	context-free	Yes
less than half	$ A \cap B < A - B $	fewer 1s than 0s	context-free	No
at least one third	$3 \cdot A \cap B \geq A $	at most three times more 0s than 1s	context-free	No
a prime number of	$ A \cap B $ is prime	a prime number of 1s	context-sensitive	No

Table 1: A list of common quantifiers with their set-theoretic definition, the string constraint instantiated by their quantifier languages, the complexity of said quantifier language, and whether the quantifier can be expressed as a morphologically simplex determiner

While the goal of Graf (2019b) was to resolve this tension, it actually exacerbates it. On the positive side, the paper showed that *an even number* is more complex than *every*, *no*, *some*, and *not all*, and it observes that among those four, *not all* differs from the three MSDQs with respect to a specific monotonicity property. But the existence of morphologically simplex *most* is still very surprising considering that its quantifier language is not even regular, let alone SL-1 or TSL-2. Following the credo that one person’s *modus ponens* is another’s *modus tollens*, Graf (2019b) presents this as additional evidence for the proposal by Hackl (2009) that *most* is built up from multiple parts and hence not an MSDQ. But this just begs the question why this option of camouflaging multiple parts as an MSDQ is unavailable for, say, *not all* or *an even number*. The account in (Graf, 2019b) thus fails to reconcile the absence of morphologically simplex *not all* with the existence of morphologically simplex *most*, in particular as the latter has a much more complex quantifier language than the former.

As I will show in Sec. 3, though, the complexity landscape changes greatly if quantifier languages do not need to be closed under permutation. While the complexity of *an even number* remains the same, *most* becomes SL-2 and now is a natural fit for the other three MSDQs. In order to fully appreciate what this means, we have to properly define what it means for a string language to be SL-2.

2.2 Strict locality over strings

Intuitively, a string language is strictly k -local (SL- k ; $k \geq 1$) iff it can be described by a finite set of permissible substrings of length k .

Example 3. Consider the string language $L := \times(10)^*\times$, which contains the strings $\times\times$, $\times10\times$, $\times1010\times$, $\times101010\times$, and so on. We can describe L in terms of five permissible bigrams: $\times\times$, $\times1$, 10 , 01 , and $0\times$. Every string in L contains only these permissible bigrams (though not necessarily all of them), and every string outside L necessarily contains at least one bigram that is not one of these five permissible bigrams. Since the permissible substrings are of length 2, L is SL-2. \lrcorner

There is an equivalent characterization of SL- k in terms of forbidden substrings (as long as $k \geq 1$), but the definition with permissible substrings will be easier to use for the purposes of this paper. More specifically, we will define SL- k in terms of positive SL- k grammars.

Given a (finite) alphabet Σ , we use Σ^* to denote the set of all possible strings over Σ , including the empty string ε , and Σ^+ for Σ^* without ε . We furthermore use Σ_E to denote $\Sigma \cup \{\times, \times\}$, where $\times, \times \notin \Sigma$ are left edge and right edge markers, respectively. For any $k \geq 1$, $\Sigma_E^k \subsetneq \Sigma_E^*$ is the set of all strings over Σ_E whose length is exactly k . If Σ contains exactly one symbol σ , then we write σ^k , σ^* , σ^+ instead of $\{\sigma\}^k$, $\{\sigma\}^*$, and $\{\sigma\}^+$, respectively. Given a string s , w is a k -factor (or k -gram) of s iff $w \in \Sigma_E^k$ and there exist (possibly empty) strings u and v over Σ_E such that $s = uvw$. We write $f_k(s)$ for the set of all k -factors of string s ; if the length of s is strictly less than k , then $f_k(s)$ is undefined.

Definition 1 (Strictly k -local). A (positive) SL- k grammar over alphabet Σ is a (possibly empty) set $G \subseteq \Sigma_E^k$. The string language generated by G is $L(G) := \{s \mid f_k(\times^{k-1}s\times^{k-1}) \subseteq G\}$. A string language L is SL- k iff there is an SL- k grammar

G such that $L(G) = L$. A string language L is strictly local (SL) iff there is some k such that L is SL- k . \lrcorner

Every strictly k -local string language is regular as it can be recognized by a deterministic finite-state automaton where each state memorizes the $k - 1$ most recent symbols. However, not every regular string language is strictly local.

Example 4. Consider the string language $L := (0^* 1 0^* 1 0^*)^*$, which consists of all strings over $\{0, 1\}$ that have an even number of 1s. This includes ε , 11, 1111, 111111, and so on. Suppose this language were strictly k -local for some even k . Then $1^k \in L$ but $1^{k+1} \notin L$. But all the k -factors of $\times^{k-1} 1^{k+1} \times^{k-1}$ (which are $\times^i 1^j$ and $1^j \times^i$ for all $i, j \geq 0$ such that $i + j = k$) are also k -factors of $\times^{k-1} 1^k \times^{k-1}$. With $k = 2$, for instance, $f_2(\times 111 \times) = \{\times 1, 11, 1 \times\} = f_2(\times 11 \times)$. Hence every strictly k -local grammar that generates $1^k \in L$ also generates $1^{k+1} \notin L$, and thus L cannot be strictly k -local. Since k was arbitrary, L is not strictly local. \lrcorner

In fact, the class SL of strictly local string languages is maximally weak in the sense that no other class has been proposed that includes infinitely many infinite languages and is properly subsumed by SL. The class SL on its own instantiates an infinite hierarchy — the class of SL- k string languages is a proper subclass of the class of SL- $(k + 1)$ languages for all $k \geq 1$. In this paper, I focus on the very bottom of this hierarchy, i.e. SL-1 and SL-2.¹ Since every SL-1 string language is also SL-2, the latter is the more important class for this paper. I argue that all MSDQs are maximally simple in the sense that they have SL-2 *verification patterns*, thus resolving the puzzle posed by *most*.

3 The verificational simplicity of *most*

We are now ready to formulate the central insight of this paper: the complexity of quantifiers can be measured in terms of their verification patterns (Sec. 3.1), and doing so reveals all attested MSDQs to form a natural class in the sense that they are SL-2 *verifiable*, which means that their verification

¹The class SL-0 can be defined but is pathological. The only possible SL-0 grammar is the empty set \emptyset . Depending on whether one interprets \emptyset as a positive grammar or a negative grammar (i.e. a set of forbidden 0-factors), it generates either the empty language or all of Σ^* . This is the only case where the generative capacity of positive and negative SL grammars diverges, which provides good reason not to include SL-0 in the definition of SL.



Figure 1: By rearranging the marbles such that there never are two white marbles next to each other, Mary can verify whether most marbles are black without counting all the marbles or calculating their relative proportions.

patterns are SL-2 string languages (Sec. 3.2). Admittedly, this hinges on defining *most* as *at least half* instead of *more than half* (Sec. 3.3), and additional restrictions are needed to rule out unattested MSDQs (Sec. 3.4). But this still marks a significant step away from the status of *most* as a complexity outlier among MSDQs.

3.1 From quantifier languages to verification patterns

The complexity results in Tab. 1 hold with respect to quantifier languages that are closed under permutation. The idea behind permutation closure is that the conditions that a quantifier $Q(A, B)$ imposes on A and B hold irrespective of what linear structure one imposes on A . From a linguistic perspective, however, this may distort the cognitive complexity of quantifiers.

Example 5. Suppose the Assistant Dean of the Office of Deranged Tasks has taken a bag with an odd number of marbles in two colors, black and white, and has meticulously arranged them in a line that spans across all the rooms of said office. Mary is then tasked by the Assistant Dean to determine whether most of the marbles are black. Mary cannot eyeball the whole line at once or rely on other heuristics. At first she considers counting, but after a long day of work she does not want to spend the mental effort required to keep track of numbers.

Instead, Mary opts for a simpler solution that does not require counting. She puts all marbles back into the bag and then builds a new line according to the following rules: The first marble must be a black, and each white marble must be immediately to the right of a black marble (see Fig. 1). If Mary ever reaches a point where these rules cannot be met, then it is not the case that most marbles are black. She happily reports her findings to the Assistant Dean, who fires her on the spot for having altered the meticulous marble arrangement. \lrcorner

While Mary in our example was under an implicit obligation to keep the order of elements undisturbed, this requirement does not hold for the interpretation of quantifiers. The complexity of L_Q

expresses how difficult it is to determine the value of $f_Q(A, B)$ given an arbitrary order of A . An alternative measure would look at how difficult it is to define a verification pattern for f_Q , i.e. a pattern that guarantees that $f_Q(A, B)$ is true iff the elements of A can be arranged according to that pattern.

Definition 2 (Verification pattern). Let L_Q be the (permutation-closed) quantifier language of some type $\langle 1, 1 \rangle$ quantifier Q . We call a set V_Q of strings over $\{0, 1\}$ a *verification pattern for Q* iff the permutation closure of V_Q is L_Q . Given a class C of string languages, we say that Q is C *verifiable* iff Q has some verification pattern V_Q in C . \square

Note that while quantifier languages are unique, a quantifier may have many distinct verification patterns, which in turn may differ in complexity.

Example 6. The set 1^+0^* is a verification pattern for *some* as its permutation closure is the set of all strings that contain at least one instance of 1. The set $0^*1^+0^*$ is also a verification pattern, but it is more complex. The verification pattern 1^+0^* is SL-2 as it is generated by the positive grammar $\{\times 1, 11, 10, 00, 1\times, 0\times\}$. But $0^*1^+0^*$ is not SL: for any choice of k , $f_k(\times^{k-1}0^k\times^{k-1}) \subseteq f_k(\times^{k-1}0^k 1 0^k\times^{k-1})$, and thus every SL- k grammar that generates $0^k 1 0^k \in 0^*1^+0^*$ also generates $0^k \notin 0^*1^+0^*$. Nonetheless, the existence of an SL-2 verification pattern for *some* entails that this quantifier is SL-2 verifiable.

As we will see next, the shift from quantifier languages to verification patterns greatly alters the complexity landscape and brings the complexity of *most* in line with other MSDQs.

3.2 SL-2 verification patterns cover the typology

The class of SL-2 string languages is extremely restricted in terms of its expressivity. For example, many phenomena in phonology are strictly local, but not all of them are strictly 2-local.

Example 7. Intervocalic voicing can be construed as a phonotactic constraint against sequences where a voiceless sound occurs immediately between two vowels. This is SL-3: the set of permissible trigrams does not contain any xyz such that x and z are vowels and y is a voiceless sound. But it is not SL-2 because, say, illicit *asola* only contains bigrams that also occur in *as* or *sola*, neither one of which violates intervocalic voicing. \square

Quantifier	1	0
$ A = 0$		
every	✓	
no		✓
always true	✓	✓

Table 2: All four SL-1 grammars over $\{1, 0\}$ and the quantifiers that they generate verification patterns for

The verification patterns for MSDQs, however, all seem to be SL-2.

Consider first the class of SL-1 string languages over $\Sigma := \{0, 1\}$. For SL-1 languages, we do not need to add edge markers to the alphabet because for $k = 1$, $\times^{k-1}s\times^{k-1} = \times^{1-1}s\times^{1-1} = \times^0s\times^0 = s$ for every string s . Hence there are only four distinct SL-1 grammars over this alphabet, each one a subset of Σ . The empty grammar allows nothing at all and generates the empty language. The grammar $\{0, 1\}$ allows everything and thus generates Σ^* . Both are pathological from a linguistic perspective. The empty language is a verification pattern for the quantifier that requires $|A| = 0$ irrespective of how B is chosen, which is unlike any generalized quantifier in natural language. Similarly, Σ^* is the verification pattern of a tautological quantifier Q with $Q(A, B) = 1$ for all A and B . The two remaining grammars are $\{1\}$ and $\{0\}$, which are more interesting. The former generates all members of 1^* , and the latter generates all members of 0^* . These are the verification patterns for *every* (without existential import) and *no*, respectively (since these verification patterns are already closed under permutation, we have $V_{\text{every}} = L_{\text{every}}$ and $V_{\text{no}} = L_{\text{no}}$). The class of SL-1 string languages over $\{0, 1\}$ thus already furnishes verification patterns for *every* and *no* (see also Tab. 2), and thus *every* and *no* are both SL-1 verifiable.

The space of SL-2 grammars is significantly larger. There are $4^2 = 16$ distinct bigrams in Σ_E . Even though 7 of them can never be members of $f_2(\times s \times)$ for any string s (e.g. $0\times$, $\times 1$, and $\times \times$), this still leaves us with 9 useful bigrams, and hence $2^9 = 512$ distinct grammars. The total number of verification patterns is smaller because some grammars generate the same string language, for instance $\{\times \times\}$ and $\{\times \times, \times 1\}$. Nonetheless the range of options is too large to discuss all of them here. Instead, I only consider grammars where strings must always start with 1 (the grammar contains $\times 1$ but not $\times 0$ or $\times \times$) and strings can end

in 1 or 0 (the grammar contains both $1\times$ and $0\times$). This leaves us with 16 distinct grammars which differ only with respect to which of the following four bigrams they contain: 11, 10, 01, 00. Surprisingly, this is enough to generate the verification patterns for all MSDQs that aren't SL-1 verifiable, including *most*.

Table 3 lists each grammar and the quantifier that corresponds to the generated verification pattern. Out of those sixteen grammars, five generate verification patterns for unnatural quantifiers: 1), 3), 4), 5), and 11). In addition, all four of 2), 7), 8) and 14) generate the same verification pattern, which is for *every* with existential import (due to the mandatory 1 at the beginning of each string). Finally, 13) and 16) generate distinct verification patterns — 1^+0^* and $1\{0,1\}^*$, respectively — but since both impose no requirements beyond the presence of at least one 1, they are both verification patterns for *some*. The remaining verification patterns are for five distinct generalized quantifiers: *all except for at most one*, *half*, *exactly one*, *at most half*, and crucially, *at least half/most*. So even though we saw in Sec. 2.1 that L_{most} is much more complex than L_{every} , L_{no} , and L_{some} , their verification patterns are of similar complexity. The property that holds of every attested MSDQ Q is that V_Q is SL-2. In other words, every attested MSDQ is SL-2 verifiable.

3.3 most = at least half?

The reader may object that the discussion so far incorrectly conflates *most* with *at least half*. The truth-conditional definition of *most* is usually given as $|A \cap B| > |A - B|$ rather than $|A \cap B| \geq |A - B|$; or equivalently, as $|A \cap B| > \frac{1}{2}|A|$ rather than $|A \cap B| \geq \frac{1}{2}|A|$. The standard definition thus equates *most* with *more than half* rather than *at least half*. There are several responses to this challenge.

First, the verification pattern identified with *at least half/most* in Tab. 3 is $1^+(01^+)^*(0)$ — the string must start with 1, may end with 1 or 0, and 1s can be followed by 1 or 0, but 0 cannot be followed by 0. If $|A|$ is odd, this pattern necessarily contains more 1s than 0. Hence the discrepancy between the verification pattern and the standard definition only arises with domains of even cardinality. But it is unclear whether the association with *at least half* rather than *more than half* is at odds with native speakers' judgments in this case. This is because native speakers generally expect *most*

to indicate that $|A \cap B|$ is noticeably larger than $|A - B|$. Hence the standard definition needs to be augmented with a mechanism such as pragmatic strengthening or a theory of vagueness in order to account for the observed behavior (see Carcassi and Szymanik 2021 for a recent discussion). Once that modification is made, though, the difference between $|A \cap B| > |A - B|$ and $|A \cap B| \geq |A - B|$ becomes immaterial.

Second, it may be the case that speakers expect verification patterns to use all bigrams in the grammar. In that case, the verification pattern for *most* will always include at least one instance of 11 and thus contain more 1s than 0s. This approach will be discussed in greater detail in Sec. 4.2.

Finally, we could consider a modified verification pattern where strings can only end in 1. This, too, would ensure that there are always more 1s than 0s, and it would not change the fact that *most* has an SL-2 verification pattern. However, this undermines one advantage of SL grammars relative to finite-state automata, namely that they can easily be viewed as generators of infinite strings as long as they impose no constraints on how a string may end. As will be discussed in Sec. 4.1, this furnishes a new way to analyze statements like “most natural numbers are not a multiple of three”, which are challenging for definitions based on cardinality. Requiring the verification pattern of *most* to both start and end with 1 thus addresses the minor mismatch in definitions over finite domains, but it does so at the cost of making it harder to work with infinite domains.

3.4 Fitting the typology

The shift from quantifier languages to verification patterns has revealed *most* to be no more complex than other quantifiers such as *some*. Quantifiers such as *a third of* or *an even number of*, on the other hand, are not SL-2 verifiable (and *an even number of* isn't even SL verifiable, cf. Example 6). This explains why *most* mirrors *every*, *no* and *some* in that at least some languages have morphologically simplex realizations of *most* whereas no such realizations are attested for *a third of* or *an even number of*. What distinguishes *every*, *no*, *some*, and *most* from *a third of* and *an even number of* is that the former are SL-2 verifiable.

SL-2 verifiability does not entail, though, that a quantifier can be an MSDQ. We already saw in Sec. 3.2 that the class of SL-2 verifiable quantifiers includes at least five highly unnatural quantifiers.

	Quantifier	11	10	01	00	Dead ends?	Useless bigrams?
1)	$ A = A \cap B = 1$					✓	
2)	every (existential import)	✓					
3)	$1 \leq A \leq 2 \ \& \ A \cap B = 1$		✓			✓	
4)	$ A = A \cap B = 1$			✓		✓	✓
5)	$ A = A \cap B = 1$				✓	✓	✓
6)	all except for at most one	✓	✓			✓	
7)	every (existential import)	✓		✓			✓
8)	every (existential import)	✓			✓		✓
9)	half (+/- 1)		✓	✓			
10)	exactly one		✓		✓		
11)	$ A = A \cap B = 1$			✓	✓	✓	✓
12)	at least half/most	✓	✓	✓			
13)	some	✓	✓		✓		
14)	every (existential import)	✓		✓	✓		✓
15)	at most half		✓	✓	✓		
16)	some	✓	✓	✓	✓		

Table 3: List of quantifiers whose verification pattern only contains strings starting with 1

This was under the additional restriction that strings must start with 1. If strings are allowed to start with 0, then many more quantifiers are SL-2 verifiable, including some that are attested but never have a morphologically simplex realization.

Example 8. The quantifier *not all* is SL-2 verifiable as its verification pattern is generated by the SL-2 grammar $\{\times 0, 00, 01, 11, 0\times, 1\times\}$. This makes *not all* a counterpart to 14) for *every* in Tab. 3 where $\times 1$ has been replaced with $\times 0$. \lrcorner

It follows that SL-2 verifiability is a necessary property but not a sufficient one.

It is tempting, then, to look for additional restrictions that prune down the set of all SL-2 verifiable quantifiers to just those that can be realized as morphologically simplex determiners. Perhaps unsurprisingly, there are multiple options that differ slightly in what set they pick out. This is illustrated in Tab. 3 for those verification patterns that must start with 1.

If every SL-2 grammar must contain 11, this rules out all unnatural quantifiers and leaves only (several versions of) *every* and *some*, as well as *most* and *all except for at most one*.

Alternatively, one could require that only the bigrams with edge markers may be *dead ends*, i.e. bigrams that make it impossible to continue the string.

Example 9. The verification pattern for *all except for at most one* is generated by the SL-2 grammar $\{\times 1, 11, 10, 1\times, 0\times\}$. Here 10 is a dead end. Once we encounter 10 in a string, we know that we have reached its end.

Now consider the minimally different SL-2

grammar $\{\times 1, 11, 10, 00, 1\times, 0\times\}$, which generates a verification pattern for *some*. Here 10 is not a dead end. If one encounters 10, it is still possible for the string to continue with an arbitrary number of 0s. The only dead ends are $1\times$ and $0\times$. \lrcorner

The intuition behind the ban against dead ends is that a verification procedure should not be at risk of getting stuck before all elements have been evaluated. This requirement rules out all unnatural quantifiers and *all except for at most one*, but leaves *half* and *exactly one*. Interestingly, *half* has a simplex realization as a noun in Russian, and depending on one’s semantic priors *exactly one* could be taken to be realized by the numeral *one*. One characterization of MSDQs, then, is as the class of SL-2 verifiable quantifiers whose verification patterns must start with 1 and whose SL-2 grammars must not contain dead ends (other than $1\times$ and $0\times$).

This is just one of many conceivable characterizations. As an illustration, Tab. 3 also indicates whether a given SL-2 grammar contains *useless* bigrams. A bigram is useless if it does not appear in any string generated by the grammar. Forbidding grammars with useless bigrams eliminates some but not all unnatural quantifiers, and it rules out several variants of *every* with existential import. This is not necessarily a good thing as it undermines some analytical options that are briefly explored in Sec. 4.5.

Yet another approach would limit the focus to just the SL-2 grammars 2) for *every*, 6) for *all except for at most one*, 12) for *most*, and 16) for *some*. These can be picked out by positing a hierarchy $11 < 10 < 01 < 00$ such that a grammar may contain a bigram y only if it also contains

all x to the left of y . Monotonicity requirements of this kind seem to be common across language modules (Keenan and Comrie, 1977; Keine, 2016; Graf, 2019a, 2020; Moradi, 2020, 2021a,b). In combination with the ban against dead ends, this monotonicity requirement would only leave *every*, *no*, *some*, and *most*, but again at the cost of losing the analytical options in Sec. 4.5.

In sum, it is certainly possible to formulate additional restrictions on SL-2 verifiability to pick out specific subclasses that more closely match the attested typology. More work is needed to determine which set of restrictions is the most elegant and insightful. Even without these restrictions, though, SL-2 verifiability provides a very tight upper bound on quantifier complexity while readily accommodating a large number of natural language D-quantifiers, including all that can have morphologically simplex realizations.

4 Exploratory remarks

4.1 Claims over infinite domains

One problem of the semantic automata approach is that it represents the domain A as a string, which must be finite. Infinite domains would require the switch to ω -automata (Perrin and Pin, 2004). This issue does not arise with SL- k grammars. Since SL- k grammars determine the well-formedness of each string based on its set of k -grams, they can be easily generalized to also generate infinite strings. An infinite binary string is a mapping s from the set \mathbb{N} of natural numbers into $\{\times, 0, 1\}$ such that $s(n) = \times$ iff $n = 0$. Infinite strings have no right edge and thus contain no right edge marker \times , but this does not matter for the SL-2 grammars in Tab. 3 because they contain both $1\times$ and $0\times$ and thus put not restrictions on the end of a string.

Interestingly, this means that the meaning of *most* generalizes immediately from finite domains to infinite domains. With respect to SL-2 verifiability, *most* states that it must be possible to arrange the elements of the domain A in such a manner that 0s are never repeated. Technically this is the case for both “most natural numbers are not a multiple of three” and “most natural numbers are a multiple of three”, but the latter requires a much greater rearrangement of elements relative to the standard order of natural numbers. Under the plausible assumption that finding such a suitable rearrangement is cognitively taxing, it is not surprising that speakers are likely to consider the former statement true

and the latter false.

4.2 Pragmatic strengthening

Quantifiers are subject to pragmatic strengthening. For example, *most* is usually interpreted as *most but not all*, presumably because the speaker could have said *all* instead. Pragmatic strengthening can be modeled as the requirement that the verification string must contain all the bigrams listed in the grammar (assuming the bigram is not useless and does not contain edge markers). Then a string like 111 would still be a verification pattern for *most*, but since it does not contain any instance of 10 or 01, it would also be infelicitous.² In terms of formal language theory, this corresponds to a step up from SL to the class of locally testable languages (McNaughton, 1974).

4.3 Modifying proportions

The proportion of 1s required by *most* can be modified by changing the locality domain. For instance, the SL-3 grammar $\{\times \times 1, \times 11, 111, 110, 101, 011, 11\times, 10\times, 01\times, 1 \times \times, 0 \times \times\}$ requires that the number of 1s is at least double that the number of 0s. This might be yet another instance of pragmatics going beyond the limits of SL-2 verifiability in order to strengthen the meaning of quantifiers. Perhaps the strategy could also be used to model vague quantifiers such as *many* and *few*.

4.4 Existential import

The analysis in Sec. 3 posits two different versions of *every*, one with existential import (with multiple options in Tab. 3), and one without (listed in Tab. 2). Existential import can be removed from the SL-2 grammar for a given quantifier by adding $\times\times$ to it. Similarly, pragmatics can add existential import by removing $\times\times$. The proper modeling of existential import has to be left to future work, but SL verifiability seems to be well-equipped to deal with the problem.

4.5 Typological frequency

Whereas *every* and *some* are common across languages, *no* and *most* are comparatively rare. This roughly matches the number of verification patterns we identified for each one of these quantifiers: 5

²This proposal requires that quantifier 6) be treated as yet another variant of *some* so that one can correctly capture the pragmatic strengthening of *some* to *some but not most/all* in cases where only one element of A is not an element of B .

for *every*, 2 for *some*, 1 for *no*, and 1 for *most*. Depending on which constraints on SL-2 grammars one adds or drops, these numbers may change significantly. Additional work is needed before a link between a quantifier’s typological frequency and its number of verification patterns can be deemed plausible, but the possibility is intriguing.

4.6 Parallels to syntax

The key difference between quantifiers languages and verification patterns is that the latter express the best case complexity of a given dependency where the linear order of symbols in the string does not introduce additional complications. This is comparable to a well-known split in computational syntax that underlies *Parikh’s theorem* (Parikh, 1966), the two-step approach (Morawietz, 2003; Mönlich, 2006), subregular syntax (Graf, 2022a,b), and also Minimalist syntax (Chomsky, 1995). They all observe that the complexity of syntactic dependencies is contingent on choices of linearization, recasting syntax as a system of fairly simple dependencies that interact with a complex system of linearization requirements.

Example 10. Consider the string language $(abc)^n$, which is regular. By moving all instances of c to the end of the string, we obtain the context-free language $(ab)^n c^n$ instead. If in addition we order all a s before all b s, the result is the tree-adjointing language $a^n b^n c^n$. Finally, if we allow every possible permutation, then we get the MIX language, which is a 2-MCFL (Salvati, 2015). Each one of these orderings represents a marked step up in complexity. ┘

Something similar may hold for quantifiers, with verification patterns capturing the underlying dependency imposed by quantifiers *modulo* the additional complications of actual verification in a given scenario.

4.7 The cognitive status of verification patterns

The parallel to syntax also highlights why verification patterns should not be equated with verification procedures. A verification procedure parses an input into a form that yields a verification pattern. As experimental results such as Lidz et al. (2011) and Kotek et al. (2015) arguably observe verification procedures, not verification patterns, it is not trivial to make any inferences from the former about the latter.

This again mirrors the situation in syntax: a given grammar formalism, say Minimalist grammars (Stabler, 1997, 2011a), has many different parsing algorithms ranging from CKY and Earley (Harkema, 2001) to recursive descent (Stabler, 2011b, 2013) and left-corner parsing (Stanojević and Stabler, 2018), which in turn must be combined with one of many conceivable linking theories in order to obtain predictions for human sentence processing (Kobele et al., 2013; Gerth, 2015; Graf et al., 2017; Lee, 2018; De Santo, 2020; Pasternak and Graf, 2021; Liu, 2023). Verification patterns provide a similarly rigorous approach to experimental findings. Instead of intuitive stories about the processing of quantifiers, we need I) a parsing algorithm that translates stimuli into strings matching a given verification pattern, and II) a rigorous linking theory that translates the operations of the parser into predictions about human behavior.

5 Conclusion

Among morphologically simplex quantifiers that are determiners (MSDQs), *most* is an outlier due to the complexity of its quantifier language. The picture painted by quantifier languages is misleading, though. If one does away with permutation closure and considers verification patterns instead, complexity is lowered significantly. All MSDQs have SL-2 verification patterns, and SL-2 grammars furnish several parameters that allow us to home in on just the class of typologically attested MSDQs. In addition, SL-2 patterns are extremely simple and also play a central role in phonology, morphology, and syntax, revealing quantifiers to be yet another facet of a very general piece of subregular machinery that drives language.

The approach presented in this paper is reasonably flexible and could possibly be extended to account for pragmatic strengthening, vague quantifiers and typological frequency effects, among other things. It is not limited to MSDQs, either. Future investigations of numerals, modals, and adverbial quantifiers might well confirm (or refute) the central status of SL verifiability in quantification and thus offer deep insights into how complex a meaning can be packed into simplex expressions.

Acknowledgements

The work carried out for this project was supported by the National Science Foundation under Grant No. BCS-1845344. I am grateful to the three anony-

mous reviewers for their very detailed feedback and suggestions.

References

- Phillip Burness, Kevin McMullin, and Jane Chandlee. 2021. Long-distance phonological processes as tier-based strictly local functions. *Glossa*, 6:1–37.
- Fausto Carcassi and Jakub Szymanik. 2021. ‘most’ vs ‘more than half’: An alternatives explanation. In *Proceedings of the Society for Computation in Linguistics*, volume 4, pages 334–343.
- Jane Chandlee. 2017. Computational locality in morphological maps. *Morphology*, 27:599–641.
- Jane Chandlee. 2022. Less is more: Reexamining assumptions through the narrow focus of subregularity. *Theoretical Linguistics*, 48:205–218.
- Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124.
- Noam Chomsky. 1959. On certain formal properties of grammars. *Information and Control*, 2:137–167.
- Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.
- Noam Chomsky and Marcel-Paul Schützenberger. 1963. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, Studies in Logic and the Foundations of Mathematics, pages 118–161. North-Holland, Amsterdam.
- Aniello De Santo. 2020. *Structure and Memory: A Computational Model of Storage, Gradience, and Priming*. Ph.D. thesis, Stony Brook University.
- Hossep Dolatian and Jeffrey Heinz. 2020. Computing and classifying reduplication with 2-way finite-state transducers. *Journal of Language Modelling*, 8:179–250.
- Sabrina Gerth. 2015. *Memory Limitations in Sentence Comprehension. A Structure-Based Complexity Metric of Processing Difficulty*. Ph.D. thesis, University of Potsdam.
- Thomas Graf. 2019a. Monotonicity as an effective theory of morphosyntactic variation. *Journal of Language Modelling*, 7:3–47.
- Thomas Graf. 2019b. A subregular bound on the complexity of lexical quantifiers. In *Proceedings of the 22nd Amsterdam Colloquium*, pages 455–464.
- Thomas Graf. 2020. Monotonicity in syntax. In *Monotonicity in Logic and Language*, volume 12564 of *Lecture Notes in Computer Science*, pages 35–53. Berlin, Heidelberg. Springer.
- 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, James Monette, and Chong Zhang. 2017. Relative clauses as a benchmark for Minimalist parsing. *Journal of Language Modelling*, 5:57–106.
- Martin Hackl. 2009. On the grammar and processing of proportional quantifiers: Most versus more than half. *Natural Language Semantics*, 17(1):63–98.
- Kenneth Hanson. 2023a. A computational perspective on the typology of agreement. Ms., Stony Brook University.
- Kenneth Hanson. 2023b. A TSL analysis of Japanese case. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2023*, pages 15–24.
- Henk Harkema. 2001. *Parsing Minimalist Languages*. Ph.D. thesis, University of California.
- 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.
- Adam Jardine and Jeffrey Heinz. 2016. Learning tier-based strictly 2-local languages. *Transactions of the ACL*, 4:87–98.
- Edward Keenan and Dag Westerståhl. 1996. Generalized quantifiers in linguistics and logic. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 837–894. North-Holland, Amsterdam.
- Edward L. Keenan and Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8:63–99.
- Stefan Keine. 2016. *Probes and Their Horizons*. Ph.D. thesis, University of Massachusetts, Amherst.
- Gregory M. Kobele, Sabrina Gerth, and John T. Hale. 2013. Memory resource allocation in top-down Minimalist parsing. In *Formal Grammar: 17th and 18th International Conferences, FG 2012, Opole, Poland, August 2012, Revised Selected Papers, FG 2013, Düsseldorf, Germany, August 2013*, pages 32–51. Berlin, Heidelberg. Springer.
- Hadas Kotek, Yasutada Sudo, and Martin Hackl. 2015. Experimental investigations of ambiguity: The case of *most*. *Natural Language Semantics*, 23:119–156.

- Dakotah Lambert and James Rogers. 2020. [Tier-based strictly local stringsets: Perspectives from model and automata theory](#). In *Proceedings of the Society for Computation and Linguistics*, volume 3, pages 330–337.
- So Young Lee. 2018. A minimalist parsing account of attachment ambiguity in English and Korean. *Journal of Cognitive Science*, 19:291–329.
- Jeff Lidz, Paul Pietroski, Tim Hunter, and Justin Halberda. 2011. Interface transparency and psychosemantics of *most*. *Natural Language Semantics*, 19:227–256.
- Lei Liu. 2023. Processing advantages of end-weight. *Proceedings of the Society for Computation in Linguistics*, 6:250–258.
- 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 (SCiL) 2021*, pages 39–50.
- Kevin McMullin. 2016. *Tier-Based Locality in Long-Distance Phonotactics: Learnability and Typology*. Ph.D. thesis, University of British Columbia.
- Kevin McMullin and Gunnar Ólafur Hansson. 2015. [Long-distance phonotactics as tier-based strictly 2-local languages](#). In *Proceedings of AMP 2014*.
- Robert McNaughton. 1974. Algebraic decision procedures for local testability. *Mathematical Systems Theory*, 8:60–76.
- Uwe Mönnich. 2006. Grammar morphisms. Ms. University of Tübingen.
- Sedigheh Moradi. 2020. [Morphosyntactic patterns follow monotonic mappings](#). In *Monotonicity in Logic and Language*, pages 147–165, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Sedigheh Moradi. 2021a. A formal restriction on gender resolution. In *All Things Morphology: Its Independence and Its Interfaces*, pages 41–54. John Benjamins, Amsterdam.
- Sedigheh Moradi. 2021b. *Monotonicity in Morphosyntax*. Ph.D. thesis, Stony Brook University.
- Frank Morawietz. 2003. *Two-Step Approaches to Natural Language Formalisms*. Walter de Gruyter, Berlin.
- Rohit Parikh. 1966. On context-free languages. *Journal of the Association for Computing Machinery*, 13:570–581.
- Robert Pasternak and Thomas Graf. 2021. [Cyclic scope and processing difficulty in a Minimalist parser](#). *Glossa*, 6:1–34.
- Dominique Perrin and Jean-Éric Pin. 2004. *Infinite Words. Automata, Semigroups, Logic and Games*. Elsevier, Amsterdam.
- Stanley Peters and Dag Westerståhl. 2006. *Quantifiers in Language and Logic*. Oxford University Press.
- Sylvain Salvati. 2015. [MIX is a 2-MCFL and the word problem in \$\mathbb{Z}^2\$ is captured by the IO and the OI hierarchies](#). *Journal of Computer and System Sciences*, 81:1252–1277.
- 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. 2011a. [Computational perspectives on Minimalism](#). In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.
- Edward P. Stabler. 2011b. Top-down recognizers for MCFGs and MGs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 39–48.
- Edward P. Stabler. 2013. [Two models of minimalist, incremental syntactic analysis](#). *Topics in Cognitive Science*, 5:611–633.
- Miloš Stanojević and Edward Stabler. 2018. A sound and complete left-corner parser for Minimalist grammars. In *Proceedings of the 8th Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 65–74.
- Shane Steinert-Threlkeld and Thomas F. Icard. 2013. [Iterating semantic automata](#). *Linguistics and Philosophy*, 36(2):151–173.
- Johan van Benthem. 1986. [Semantic automata](#). In *Essays in Logical Semantics*, pages 151–176. Springer, Dordrecht.

Reassessing a model of syntactic island acquisition

Avni Gulrajani

Department of Linguistics
1401 Marie Mount Hall
University of Maryland
College Park, MD 20742
avni@terpmail.umd.edu

Jeffrey Lidz

Department of Linguistics
1401 Marie Mount Hall
University of Maryland
College Park, MD 20742
jlidz@umd.edu

Abstract

This paper examines the limits of the learning model for syntactic islands from Pearl and Sprouse (2013), which challenges linguistic nativist perspectives by suggesting that island effects can be learned from language input and domain-general or learned abilities. Our investigations focus on sentences that would be ambiguous if there were no island constraints, where one conceivable interpretation violates an island constraint. A learner without any knowledge of islands could incorrectly treat the island-violating parses of such sentences as grammatical. We conducted simulations introducing these sentences in the model’s input and also analyzed their frequency in the child-directed speech corpora used as the model’s input. The results show that a small number of potentially island-violating sentences in the model’s input impairs its ability to exhibit island effects, and potential island violations occur frequently enough in children’s input to degrade the model’s performance.

1 Introduction

Island effects have played a central role in controversies around nativism in linguistics. While many linguists have argued that they are entirely a consequence of innate linguistic knowledge, Pearl and Sprouse (2013) offer a different viewpoint. They developed a computational model that suggests that these effects can be learned through language input and various abilities which might be learned or domain-general, such as parsing sentences and calculating probabilities. This model warrants thorough scrutiny as it represents the first serious attempt to explain how knowledge of islands could possibly be learned. Understanding the limitations of this model could be helpful in developing improved models of the acquisition of islands, potentially leading to a more comprehensive understanding of islands overall.

This paper explores the limits of Pearl and Sprouse’s model through computational simulations and an examination of children’s linguistic input. It specifically focuses on how different assumptions about the learner’s intake might affect the model’s performance. Originally, Pearl and Sprouse tested their model with adult-like parses of sentences. Our analysis considers the possibility that learners could misparse sentences that would be ambiguous if there were no island constraints.

To understand this issue, consider the sentences in (1). Sentence (1a) is ambiguous because the *wh*-phrase could relate to either verb, leading to different interpretations about thinking or smiling. In contrast, sentence (1b) only allows the interpretation where “why” is associated with “wonder” because an island structure blocks the alternative interpretation. But a learner without any knowledge of islands might not know this about sentence (1b) and could misparse it in a way where the *wh*-phrase relates to the verb inside the island.

- (1) a. Why does Leo think that Meredith smiles?
- b. Why does Leo wonder whether Meredith smiles?

Throughout the rest of this paper, we will use the term “potential island violation” for a sentence like (1b), which is unambiguous in English but would be ambiguous if English had no islands.

Our results indicate that a very small number of potential island violations in the model’s input hinders its ability to display island effects, and children’s input contains a large enough number of these sentences to degrade the model’s performance.

2 A description of syntactic islands

Languages allow certain dependencies to extend over any number of words or phrases; however, these dependencies can still be restricted by partic-

ular structures. The examples in (2) demonstrate that wh-dependencies can span many clauses, but example (3) shows that the same type of dependency cannot cross even a single wh-clause. In these examples and subsequent examples, the underscore represents the position associated with the wh-phrase (called the gap position).

- (2) a. What does Meredith like ___?
 b. What does Leo think that Meredith likes ___?
 c. What does the teacher believe... that Leo thinks that Meredith likes ___?
- (3) * What does Leo wonder why Meredith likes ___?

The structures that constrain these dependencies are called syntactic islands (Ross, 1967). Many types of structures have been identified as islands, including complex noun phrases, subjects, coordinate structures, adjuncts, and wh-clauses. Examples of these are shown in (4)-(8). The island structure in each example is shown in brackets.

- (4) Complex NP: * What did he make [the claim that the teacher celebrated ___]?
- (5) Subject: * What do [pictures of ___] make you happy?
- (6) Coordinate structure: * What did she see [the elephant and ___]?
- (7) Adjunct: * What did you smile [after she said ___]?
- (8) Wh-clause * What did you ask [why she said ___]?

While these examples focus on wh-dependencies, islands also affect other kinds of dependencies, including tough movement, relative clauses, comparative deletion, and clefting. (Chomsky, 1977; Bresnan, 1975).

Many attempts have been made to create general theories explaining a variety of island effects. These theories vary, with some attributing islands to grammatical knowledge and others to factors like pragmatics of questions or sentence processing difficulties. Among the grammatical theories, one particularly noteworthy example is the Subjacency Condition (Chomsky, 1973), which restricts dependencies to positions separated by no more than one bounding node. A paraphrased version of its original definition is given in (9).

- (9) Subjacency Condition:
 No rule can involve X and Y in the structure:
 ... X ... [a ... [b ... Y ...] ...] ...
 where a and b are bounding nodes.

Often, island phenomena are used to support

linguistic nativist perspectives because comprehensive theories of islands are stated in terms of highly abstract linguistic properties which are not directly observable to learners. Island structures and island-sensitive dependencies vary widely in their surface-level characteristics, which makes them difficult to explain using directly observable properties. However, a potential concern with such abstract theories is the learning puzzle they present. Learners must somehow converge on the same abstract representations even though many representations can be compatible with their experience (cf. Chomsky, 1975, Goodman, 1955). Nativist theories address the puzzle of acquiring such abstract knowledge by considering it a component of an innate language faculty.

3 Pearl and Sprouse's model

Contrasting with theories that attribute island effects mostly or entirely to innate linguistic knowledge, Pearl and Sprouse (2013) suggest that a substantial portion of the knowledge resulting in island effects can be learned through experience. Instead of relying on innate linguistic knowledge, their model requires several biases that are possibly either learned and domain-specific, or innate and domain-general. Since linguistic nativism depends on biases that are both innate and domain-specific at once, their model could possibly challenge this perspective.

3.1 The learning process

At the beginning of the learning process, the learner is able to identify wh-dependencies, which means knowing that a wh-phrase must correspond to a gap elsewhere in the sentence. When a sentence with a wh-dependency is encountered, the learner parses the sentence into a phrase structure tree and extracts a sequence of "container nodes," which are phrasal nodes in the tree that contain the gap but not the wh-phrase. While parsing sentences, CP nodes are subcategorized according to the lexical item that introduces the CP. Next, the sequence of container nodes is broken into smaller sequences of three container nodes, called trigrams. The learner records the individual frequencies of trigrams and the total number of trigrams observed throughout a period of time. A small smoothing constant of 0.5 is added to all trigram frequencies, so even unobserved trigrams have a frequency of 0.5.

A "grammaticality preference" for a sentence

is calculated by multiplying the probabilities of all trigrams in its container node sequence. The probability of a trigram is estimated by dividing its frequency by the total number of observed trigrams.

Below is a walk-through of the process of learning and calculating a grammaticality preference, demonstrated with a specific sentence example.¹

(10) Sentence: What do you think she saw?

Parsed sentence:

[CP What do [IP [NP you] [VP think [CP
[IP [NP she] [VP saw ____]]]]]]]

Container node sequence:

IP–VP–CP_{null}–IP–VP

Trigrams:

start–IP–VP

IP–VP–CP_{null}

VP–CP_{null}–IP

CP_{null}–IP–VP

IP–VP–end

Updating trigram counts:

add 1 each trigram count

add 5 to the number of trigrams observed

Calculating a grammaticality preference:

Grammaticality preference =

$P(\text{start-IP-Vp}) \times P(\text{IP-Vp-CP}_{\text{null}}) \times$

$P(\text{VP-CP}_{\text{null-IP}}) \times P(\text{CP}_{\text{null-IP-Vp}}) \times$

$P(\text{IP-Vp-end})$

These learning biases enable the learner to generalize beyond the input while still avoiding ungrammatical sentences. Focusing exclusively on wh-dependencies and container node sequences ensures that the learner avoids learning from irrelevant information. Subcategorizing CPs is a necessary step in distinguishing certain island violations from grammatical sentences. Without this information, *whether* and adjunct island violations, which are characterized by CP_{whether} and CP_{if} nodes, would be indistinguishable from grammatical dependencies that include CP_{that} or CP_{null} nodes. Keeping track of trigram probabilities and calculating the grammaticality of a dependency from the probabilities of its trigrams allows the learner’s knowledge to extend beyond the specific sentences that have been observed. If a new sentence has a dependency containing frequent trigrams, it is perceived as grammatical even if the whole sentence or container node sequence has never been encountered before. Pearl and Sprouse note that although these biases are conducive to learning, some of

¹Grammaticality preferences are not necessarily calculated after each sentence observation, but the calculation process is included here for clarity.

them have no other obvious motivation. It’s not obvious that a learner would know to pay close attention to small sequences of nodes involved in wh-dependencies without any prior knowledge that islands exist. Still, this model is important because it appears to demonstrate the possibility of acquiring knowledge of islands without innate island constraints.

3.2 The model’s input

The input for the model consists of 200,000 container node sequences, randomly selected from a frequency distribution that represents approximately 21,000 wh-dependencies from four child-directed speech corpora: the Adam and Eve corpora from the Brown dataset (Brown, 1973), the Valian corpus (Valian, 1991), and the Suppes corpus (Suppes, 1974). The number 200,000 is Pearl and Sprouse’s estimate of the number of wh-dependencies a child would encounter between the ages of 2 and 5. According to Pearl and Sprouse, this period spans the time from when children start recognizing wh-dependencies to when they exhibit knowledge of islands.

3.3 Measuring the success of the model

Pearl and Sprouse compared the model’s grammaticality preferences to adult acceptability judgements in experiments from Sprouse et al. (2012). Here, island effects were defined as superadditive interactions between two factors: gap position (MATRIX or EMBEDDED) and structure (ISLAND or NON-ISLAND). Example (11) includes different combinations of gap position and structure for *whether* islands. The interaction is measured using the differences-in-differences score, which is calculated by subtracting the difference in the MATRIX conditions from the difference in the EMBEDDED conditions.

- (11) a. MATRIX | NON-ISLAND: Who ____ thinks that Leo plays piano?
 b. EMBEDDED | NON-ISLAND: What does Meredith think that Leo plays ____?
 c. MATRIX | ISLAND: Who ____ wonders whether Leo plays piano?
 d. EMBEDDED | ISLAND: * What does Meredith wonder whether Leo plays ____?

In addition to *whether* islands, Sprouse et al. also tested complex NP islands, subject islands, and adjunct islands. The results of these experiments show superadditive interactions for all four

Island type	MATRIX NON-ISLAND	EMBEDDED NON-ISLAND	MATRIX ISLAND	EMBEDDED ISLAND	Differences-in- differences
Subject	-1.21	-7.89	-1.21	-20.17	12.28
Complex NP	-1.21	-13.84	-1.21	-19.81	5.97
Whether	-1.21	-13.84	-1.21	-18.54	4.7
Adjunct	-1.21	-13.84	-1.21	-18.54	4.7

Table 1: Model’s grammaticality preferences and differences-in-differences for four island types. To maintain consistency with Pearl and Sprouse’s reported results, all values in this table are presented as log probabilities.

island types. Similarly, Pearl and Sprouse tested their model on the same sentence types and found superadditive patterns in the model’s grammaticality preference scores for all island types tested. These scores and their differences-in-differences are shown in Table 1.

4 Interpreting the model’s results

Before examining the model’s response to potential island violations, it is important to clarify the extent of its success to begin with. Although the original results demonstrate the model’s success at displaying island effects for four specific island types, it remains unclear whether it achieves a true separation of island structures from all other structures.

The model’s probability-based grammaticality preference scores are used as replacements for both acceptability and grammaticality at once, although the exact relationships between these concepts are not straightforward (see Phillips, 2013 for discussion). Since the model is not designed to encompass all aspects of acceptability judgements, there are noticeable differences between its scores and true acceptability judgements. For example, experiments from Sprouse et al. (2012) show that the presence of an island structure outside a wh-dependency affects acceptability, but the model does not display this pattern because it ignores all properties of a sentence other than the nodes in its dependency. This might be appropriate if the model is only supposed to detect differences in grammaticality; however, the model also seems to capture some acceptability judgement patterns that go beyond grammaticality alone, such as the effect of a dependency’s length. In general, the model assigns lower scores to longer dependencies because it involves multiplying many probabilities between 0 and 1.²

²The model’s preference for shorter dependencies might initially seem desirable, since acceptability judgements share this pattern. However, the underlying reasons for these preferences are quite different. Long dependencies are rated as less acceptable because of parsing difficulties that are unrelated

Since it is unclear which exact components of acceptability judgements the model’s scores are supposed to represent, it could be more productive to focus on the broader idea that learning to identify islands involves separating them from all other structures in some way. If there is a detectable pattern in the input that distinguishes islands from non-island structures, then the model’s scores should reflect this distinction somehow, regardless of how exactly they relate to acceptability and grammaticality. According to Pearl and Sprouse, the definition of an island effect is a superadditive pattern. So, islands should be associated with stronger superadditive patterns than non-island structures if the model is successful.

Using this definition, the model does not achieve a perfect separation of islands from other structures. Superadditive patterns appear even when comparing sentences without island violations, suggesting that this measure is susceptible to false positives. Because the model is unaffected by island structures outside of wh-dependencies, the differences-in-differences measurement effectively reduces to a single difference, and a superadditive pattern appears with any difference at all between two probabilities. Since the model prefers shorter dependencies, and trigram probabilities naturally vary widely, these differences appear in nearly any pair of sentences compared. Table 2 presents a variety of similarly acceptable sentence pairs whose differences in grammaticality preference scores exceed those associated with island violations.³ Although it might be impractically difficult to create a complete model of acceptability judgements, verifying the model’s success still requires an explanation of why its superadditive effects are relevant in situations involving island violations but not in others. Without this explanation, it seems that the model

to probability (Gibson, 1998; Sprouse, 2020). By attributing these low ratings entirely to probability, the model possibly overestimates the impact of probability on acceptability.

³Although we haven’t run experiments showing that these sentences are similar in acceptability, it seems unlikely that they would show differences as large as true island effects.

Sentence 1	Sentence 2	Difference in grammaticality preferences
What did she think he saw? IP-VP-CP _{null} -IP-VP	What did she think that he saw? IP-VP-CP _{that} -IP-VP	6.61
What did she think about? IP-VP-PP	What did she think about seeing? IP-VP-PP-IP-VP	10.43
What did she see? IP-VP	What did she see a picture of? IP-VP-NP-PP	9.93
What was she hoping to see? IP-VP-IP-VP	What was she happy to see? IP-VP-AdjP-IP-VP	14.77
What did she want him to see? IP-VP-IP-VP	What did she hope for him to see? IP-VP-CP _{for} -IP-VP	11.08
What did she allow him to see? IP-VP-IP-VP	What did she give him a chance to see? IP-VP-NP-IP-VP	11.64
What did she think he saw? IP-VP-CP _{null} -IP-VP	What did she feel like he saw? IP-VP-PP-CP _{null} -IP-VP	7.83

Table 2: Differences in log probabilities of similarly acceptable sentences. Below each sentence is its container node sequence.

cannot easily distinguish between these.

4.1 Unobserved trigrams

It is possible that slight adjustments to the learning procedure could result in a clearer separation of islands from other structures. Pearl and Sprouse mention an important distinction between island violations and grammatical dependencies: island violations always contain at least one trigram that has never been observed, whereas grammatical dependencies consist of trigrams that have been observed previously, even if infrequently. To differentiate these cases, they suggest calculating grammaticality preferences in ways that penalize unseen trigrams more strongly. For example, instead of taking the product of trigram probabilities, grammaticality preferences could be calculated using the geometric mean instead, which moderates the impact of multiplying many probabilities. Another possible solution is to lower the smoothing constant to a much smaller number, which further decreases the probabilities of unobserved trigrams, and consequently any dependencies containing these trigrams. A third idea is that the learner could “simply note the presence of a very low-probability trigram,” instead of aggregating trigram probabilities.

However, a potential remaining problem with all of these suggestions is that they all depend on island violations containing unobserved trigrams. If the model’s input includes even a single island violation, the model could still fail to differentiate the island violation from other rare grammatical dependencies even after employing these strategies. This is particularly likely if potentially island-violating sentences are parsed incorrectly. The next section focuses on this issue.

5 Addressing potential island violations

We explored the impacts of potential island violations in the model’s input using two approaches. First, we conducted simulations where we incorporated varying numbers of possibly island-violating sentences in the model’s input, regardless of their presence in children’s actual input. The purpose of these simulations was to assess the model’s capacity to handle potential island violations and identify the number of potential island violations that would cause it to be unable to display island effects. Second, we searched through the four child-directed speech corpora used as input for potential island violations and included their island-violating parses in the model’s input. This analysis was intended to determine the frequency of potential island violations in children’s input and whether a model with limited tolerance for island violations could still succeed.

In both of these investigations, it was necessary to modify the model’s process for selecting input container node sequences so that it could accommodate ambiguous sentences. Originally, each sentence was represented by a single container node sequence, and 200,000 sequences were randomly chosen one at a time from this collection. In our new setup, sentences are represented as groups of container node sequences, and the selection procedure involves selecting a sentence and one of its possible container node sequences randomly, meaning each parse for a particular sentence has an equal chance of being selected. This might overestimate the chance that a learner would misparse potential island violations, but we want to consider the worst-case scenario to understand the full range

of possibilities (contrasting with Pearl and Sprouse, who focused on the best-case scenario). In the absolute worst case, learners would consistently choose island-violating parses, but this situation seems unlikely. Instead, we are considering a more realistic worst-case scenario where learners are completely unbiased.

5.1 Simulations

For each island type, we attempted to include potential island violations with the exact EMBEDDED | ISLAND container node sequences used by Pearl and Sprouse. This regime required sentences with adjunct wh-phrases and verbs inside island structures. Consequently, it was possible to find such sentences for all island types except subject islands, which are typically nominal. Examples of the types of potential island violations identified are shown in (12), (13), and (14), along with the container node sequences of the island-violating parses.

(12) Complex NP island:

Why did Meredith make the claim that Leo plays piano?

Grammatical: IP-VP

Island-violating: IP-VP-NP-CP_{that}-IP-VP

(13) *Whether* island:

Why does Meredith wonder whether Leo plays piano?

Grammatical: IP-VP

Island-violating: IP-VP-CP_{whether}-IP-VP

(14) Adjunct island:

How does Meredith smile if Leo plays piano?

Grammatical: IP-VP

Island-violating: IP-VP-CP_{if}-IP-VP

We conducted a separate simulation for each island type and examined the model’s grammaticality preference scores for each pair of EMBEDDED | ISLAND and EMBEDDED | NON-ISLAND sentences after including different numbers of island-violating parses. We ignored the MATRIX gap position conditions because the model always rates them as the same. The EMBEDDED | NON-ISLAND baseline for these three island types is IP-VP-CP_{that}-IP-VP. The results are displayed in Figure 1 and explained below.

For *whether* islands and adjunct islands, including just five island violations of each type results in higher scores for island-violating sentences than the grammatical baseline. Complex NP island effects might better withstand island violations in the input for two reasons. First, the grammatical sentence has an advantage because of its shorter con-

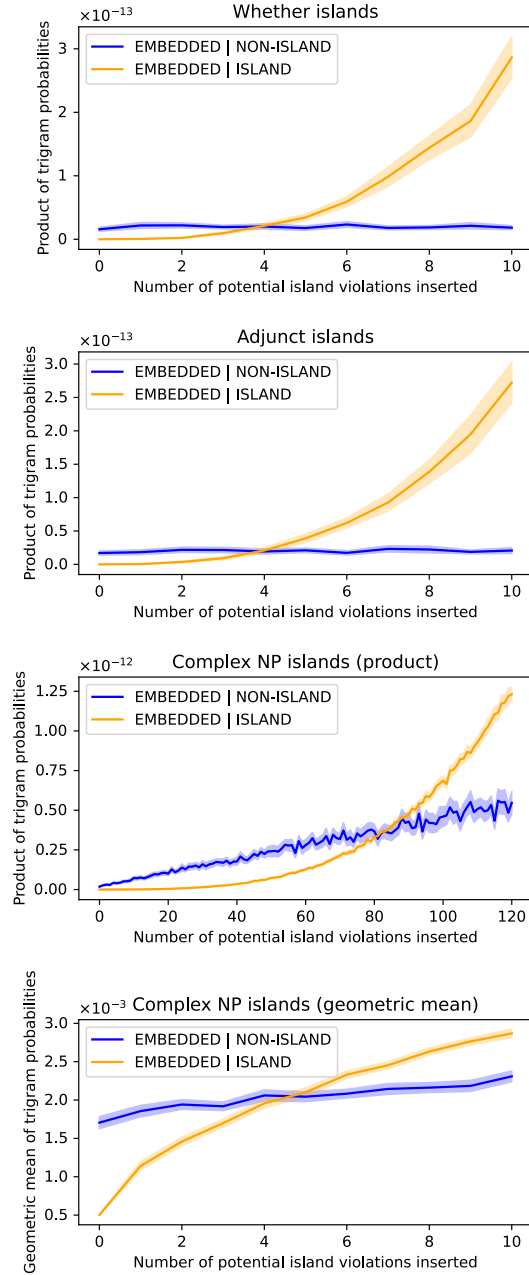


Figure 1: Grammaticality preferences with varying numbers of island violations in the input. Each pair of points represents the average of 50 repetitions of the model. Colored areas represent 95% confidence intervals. These charts display raw probabilities instead of log probabilities for clearer visualization.

tainer node sequence. Second, the two container node sequences share many trigrams, so observing island violations actually increases the score of the grammatical sentence. As a result of these two issues, this island effect is quite persistent; it remains until approximately 90 potential island violations are inserted. However, when using alternative mea-

surements that do not favor shorter dependencies, like the geometric mean, the complex NP island effect disappears with just five potential island violations.

While these simulations clearly demonstrate that *whether* and adjunct island effects disappear with a small number of island violations in the input, interpreting the results for complex NP islands depends heavily on the exact method used to calculate grammaticality preferences. Using Pearl and Sprouse’s original approach, it might seem like a small number of island violations has no serious impact on the complex NP island effect. However, as explained in Section 4, this approach leads to difficulties in differentiating between true island violations and uncommon grammatical dependencies. To achieve a clearer separation, several solutions were suggested which all focus on penalizing unseen trigrams, since this is the only unique characteristic of island violations that this model can detect. Because these solutions depend on island violations containing unseen trigrams, introducing even very few potential island violations lands us back at the original problem. For this reason, even a small number of island violations in the input might present problems for the model overall.

The reason only five potential island violations are required to eliminate these island effects is because the baseline grammatical sequence includes a rare container node, CP_{that} , which only appears twice in the entire input corpus. We selected this container node sequence to remain consistent with Pearl and Sprouse’s original tests, but it’s worth considering what might have happened if we had used a more common baseline, such as one with CP_{null} . In this situation, more potential island violations would be required to undo the island effects, but the challenge of distinguishing island violations from rare grammatical dependencies would remain the same.

5.2 Children’s input

After examining the child-directed speech corpus used as the model’s input, we found several different types of potential island violations, presented in Table 3.

We included the island-violating container node sequences for each potential island violation and retested the model with this revised input. We tested various island types, including two of the four types tested by Pearl and Sprouse, excluding subject and *whether* islands because of their ab-

Island type	Example sentence	Count
Complex NP	Adam, how would I know that those are the wheels that go on here?	24
Adjunct	How can he sit comfortably if you take all the pillows off?	69
Wh	How do you know what we find at the carnival?	35
Extraction from NP	What do you build a ship with?	68
Coordinate structure	How can the tiger be so healthy and fly like a kite?	151

Table 3: Types and frequencies of potential island violations in children’s input, with examples from the input corpus

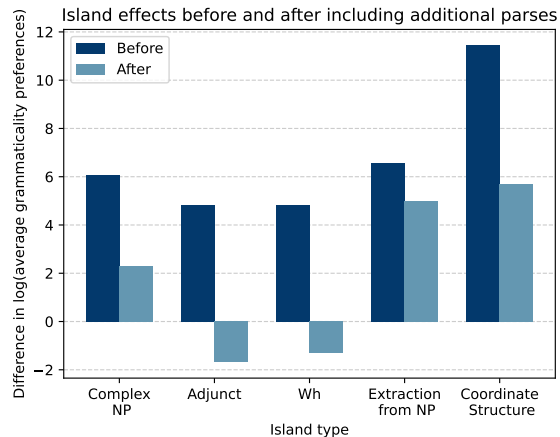


Figure 2: Differences in average grammaticality preferences (transformed to log probabilities) before and after inserting island-violating parses from children’s input. Each bar represents the result from 1,000 model runs.

sence in the input corpus. Each test involved a single comparison of an island violation and a similar grammatical sentence. The complete list of test sentences is shown in Table 4. Figure 2 displays the model’s grammaticality preferences for these test sentences before and after inserting the island-violating parses. These results indicate that the potential island violations in children’s input can impair the model’s ability to recognize several island types, although some island effects remain.

The results for complex NP and adjunct islands are consistent with the simulation results presented earlier. The island effect for adjuncts beginning with “if” disappears entirely because the input contains many instances of these. Adjuncts beginning with “when,” “while,” and “so” are similarly af-

Island type	Non-island sentence	Island sentence
Complex NP	What did he claim that she saw? IP-VP-CP _{that} -IP-VP	What did he make the claim that she saw? IP-VP-NP-CP _{that} -IP-VP
Adjunct	What did he think that she saw? IP-VP-CP _{that} -IP-VP	What did he worry if she saw? IP-VP-CP _{if} -IP-VP
Wh	What did he think that she saw? IP-VP-CP _{that} -IP-VP	What did he wonder when she saw? IP-VP-CP _{when} -IP-VP
Extraction from NP	What did he see with? IP-VP-PP	What did he see the elephant with? IP-VP-NP-PP
Coordinate structure	What did he see? IP-VP	What did he see an elephant and hear? IP-VP-VP

Table 4: Test sentences and container node sequences

ected. Other adjuncts exhibit small island effects with score differences less than 2. The complex NP island effect only partially remains. The score difference decreases to 2.31, which is smaller than many differences found between grammatical sentences.

Similar to adjuncts, wh-islands are also affected, but not uniformly. Because CPs are subcategorized by their initial words, certain wh-words form islands while others do not.⁴ Our test sentence contains an embedded clause beginning with “when,” which appears often enough in the input that the model does not consider it an island. However, embedded “why” questions are rare, so the model still treats these as islands.

Some extractions from NPs are grammatical while others are not, as shown by the examples in (15), and linguists have not conclusively determined the underlying distinctions between these (Davies and Dubinsky, 2003).

- (15) a. What did you see [a picture of ___]?
b. * What did you see [the elephant with ___]?

Since they share identical container node sequences, the model is unable to differentiate between grammatical and ungrammatical extractions from NPs and instead generally rates them low because they contain uncommon trigrams. This effect partially remains despite many potential ungrammatical extractions from NPs in the input, although its size is smaller than some differences between grammatical sentences. If the model were enhanced in such a way that it could differentiate between grammatical and ungrammatical extractions

⁴It’s not entirely clear that this is how CPs are subcategorized. According to Pearl and Sprouse, the category depends on the word that “introduces” the CP, which could mean either the complementizer or the first word. However, using the complementizer would cause the model to fail to recognize all wh-islands because wh-words are not complementizers.

tions from NPs, potential island violations could become problematic. There are 68 potential ungrammatical extractions from NPs compared to only 8 grammatical ones. The larger number of ungrammatical extractions suggests that they could interfere with learning.

The model’s ability to recognize coordinate structure island violations is uncertain to begin with. Although our test shows a large difference for this island type, the probability of a coordinate structure island violation even before adding island violations to the input is higher than that of many grammatical dependencies, such as two-clause dependencies. The difference in our test sentences partially remains after inserting island violations, probably because the baseline container node sequence is shorter and overlaps with the island-violating sequence, similar to complex NP islands and extractions from NPs. However, its size diminishes to a value smaller than some grammatical sentences display. Every grammatical sentence pair in Table 2 from Section 4 exhibits a larger score difference.

In summary, the impact of incorporating island-violating parses varies: certain adjunct and wh-island effects disappear entirely; complex NP, coordinate structure, extraction from NP, and other adjunct and wh-island effects are substantially reduced; and subjects and whether-clauses continue to display island effects. It is important to recognize that these tests were conducted using Pearl and Sprouse’s original method for calculating grammaticality preferences. If we had used alternative approaches, particularly ones that focus on unobserved trigrams, any potential island violations would have removed the island effects entirely. In this situation, only subject and *whether* island effects would remain, because only these islands contain unobserved trigrams.

6 Conclusion

This paper has concentrated on exploring the limits of Pearl and Sprouse’s model, focusing on sentences with potential island violations. Undertaking this analysis is important because their model represents a serious effort to explain how knowledge of islands could be learned from experience. Two potential problems have been identified here: the challenge of distinguishing true island violations from grammatical dependencies with low probabilities, and the possibility that sentences with potential island violations could be misparsed. Resolving these issues is important for a comprehensive understanding of island acquisition.

Of course, our simulations reflect a kind of worst case scenario by treating each potential island violation as though each parse had an equal chance of being selected. It may be that the impact of these sentences could be reduced by semantic and pragmatic factors. For example, we can imagine a learning scenario in which the child uses the discourse context to estimate the intended interpretation independent of the parse. Such a child could then use that interpretive estimate as a factor in deciding on a parse, possibly lessening the impact of the potential island violations.

It is also worth noting that the majority of the potential island violations come from adjunct questions, where there is not an independent source (such as argument structure) to identify the extraction site. It could be that learners down-weight evidence from adjunct questions precisely because they lack an independent means of verifying the extraction site. We can also imagine an enriched version of the Pearl and Sprouse model that tracks extraction paths separately for argument wh-phrases and adjunct wh-phrases. Such a model could also down-weight evidence from extraction paths that only occur for adjunct wh-phrases, on the assumption that the locality domains for adjunct wh-phrase should not be less restrictive than the locality domains for argument wh-phrases. Of course, such a model would be quite distinct in spirit from the original Pearl and Sprouse model.

Acknowledgements

We are grateful to Lisa Pearl for providing the code for the model and for her invaluable comments and feedback on this project.

References

- Joan W. Bresnan. 1975. [Comparative deletion and constraints on transformations](#). *University of Massachusetts Occasional Papers in Linguistics*, 1:Article 4.
- Roger Brown. 1973. *A First Language: The Early Stages*. Harvard University Press, Cambridge, MA.
- Noam Chomsky. 1973. Conditions on transformations. In Stephen R. Anderson and Paul Kiparsky, editors, *A Festschrift for Morris Halle*, pages 232–286. Holt, Rinehart and Winston, New York.
- Noam Chomsky. 1975. *Reflections on Language*. Pantheon Books, New York.
- Noam Chomsky. 1977. On wh-movement. In Peter W. Cullicover, Thomas Wasow, and Adrian Akmajian, editors, *Formal Syntax*, pages 71–132. Academic Press, New York.
- William D. Davies and Stanley Dubinsky. 2003. [On extraction from NPs](#). *Natural Language & Linguistic Theory*, 21(1):1–37.
- Edward Gibson. 1998. [Linguistic complexity: Locality of syntactic dependencies](#). *Cognition*, 68:1–76.
- Nelson Goodman. 1955. *Fact, Fiction, and Forecast*. Harvard University Press, Cambridge, MA.
- Lisa Pearl and Jon Sprouse. 2013. [Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem](#). *Language Acquisition*, 20(1):23–68.
- Colin Phillips. 2013. [On the nature of island constraints ii: Language learning and innateness](#). In Jon Sprouse and Norbert Hornstein, editors, *Experimental syntax and island effects*, pages 132–158. Cambridge University Press, Cambridge.
- John Robert Ross. 1967. *Constraints on Variables in Syntax*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.
- Jon Sprouse. 2020. [A user’s view of the validity of acceptability judgments as evidence for syntactic theories](#). In Samuel Schindler, Anna Drożdżowicz, and Karen Brøcker, editors, *Linguistic Intuitions: Evidence and Method*. Oxford University Press, Oxford.
- Jon Sprouse, Matthew Wagers, and Colin Phillips. 2012. [A test of the relation between working-memory capacity and syntactic island effects](#). *Language*, 88:82–123.
- Patrick Suppes. 1974. [The semantics of children’s language](#). *American Psychologist*, 29:103–114.
- Virginia Valian. 1991. [Syntactic subjects in the early speech of American and Italian children](#). *Cognition*, 40:21–81.

BERT’s Insights Into the English Dative and Genitive Alternations

Qing Yao

College of Creative Studies
& Department of Linguistics,
University of California Santa Barbara
qyao@ucsb.edu

Simon Todd

Department of Linguistics,
University of California Santa Barbara
& NZILBB, University of Canterbury
sjtodd@ucsb.edu

Abstract

We construct two models that encode varying degrees of context to predict noun phrase order in English dative constructions from their BERT embeddings. The models can successfully predict dative alternations, even without access to context. They are sensitive to features such as animacy, definiteness, and pronominality, suggesting that BERT embeddings encode such information. The best-performing model also shows reasonable success in zero-shot transfer to predicting genitive alternations, indicating some understanding of the shared factors that shape the two alternations. However, the effects of features on the transfer results are not always consistent with known influences on genitive alternations, suggesting that the model may also be drawing from other information encoded in BERT’s embeddings. These findings provide insights into the extent to which BERT exhibits human-like word order preferences and demonstrate the potential application of large language models in replacing hand-annotated features for corpus-based studies of syntactic knowledge.

1 Introduction

In the literature on language and cognition, much attention has been paid to syntactic alternations: situations where language users have an apparent choice between two ways of putting together the same words without radically altering meaning. Two such situations that have gained prominence are the English dative (Bresnan et al., 2007; Bresnan and Ford, 2010; Gropen et al., 1989; Theijssen et al., 2013) and genitive (Rosenbach, 2014; Szmrecsanyi et al., 2017; Szmrecsanyi and Hinrichs, 2008) alternations, exemplified in (1) and (2).

(1) Dative alternation

- a. **NP-dative:** Bob gives [Alice]_{recipient} [the money]_{theme}
- b. **PP-dative:** Bob gives [the money]_{theme} to [Alice]_{recipient}

(2) Genitive alternation

- a. **s-genitive:** [a car]_{possessor}’s [tires]_{possessum} are very durable
- b. **of-genitive:** [the tires]_{possessum} of [a car]_{possessor} are very durable

In this paper, we study the processing of the dative alternation in BERT (Devlin et al., 2019), in two ways. First, we ask whether pre-trained BERT embeddings can be used to predict alternant choice in dative constructions in a corpus of New Zealand English. We compare models based on BERT embeddings with different degrees of context to a model based on the array of features identified as relevant in the linguistic literature, and find that all models are similarly successful, showing that BERT embeddings encode information that is relevant to the dative alternation. Second, we use the BERT embeddings to assess how the underpinnings of the dative alternation may relate to that of the genitive alternation, by asking how well a model trained to predict the dative alternation can be zero-shot transferred to predict the genitive alternation. The degree to which transfer is possible reflects the degree to which the two alternations are shaped by shared factors, including both general-purpose considerations such as accessibility and construction-specific considerations that are paralleled between them (Diessel, 2020).

Studying the dative and genitive alternations through the lens of BERT has both theoretical and practical implications. On the theoretical side, it can help us to model the cognitive basis of probabilistic sentence production and processing preferences, including the extent to which such preferences are construction-specific and how they can be learned in a highly general way. On the practical side, it can allow us to assess the potential of using large language models to replace time-consuming hand-annotation of features for corpus-based studies of syntactic knowledge.

2 Background

2.1 The dative and genitive alternations

The dative and genitive alternations have figured into many proposals about the nature of the cognitive representations and processes that underpin syntactic knowledge, production, and processing. For example, rule- (Gropen et al., 1989) and construction-based approaches (Gries and Stefanowitsch, 2004) to the dative alternation have appealed to subtle differences in meanings represented by the verb in each alternant, giving cognitive representations of lexical semantics a central role. At the other extreme, accessibility-based approaches (Bock, 1982; MacDonald, 2013) have appealed to the cognitive bottleneck of serial lexical retrieval and highlighted a tendency to prefer alternants that order easily-retrieved arguments first, thus downplaying the role of the precise nature of representations in comparison to general information-processing constraints. In recent years, corpus, experimental, and modeling investigations (Bresnan, 2007; Bresnan and Ford, 2010; Theijssen et al., 2013) have generally supported a middle ground, in which syntactic production and processing are seen as probabilistic, influenced by an array of features including both lexical semantics and determinants of accessibility.

Extensive work has been done in understanding what factors drive these alternations (Bresnan et al., 2007; Rosenbach, 2014; Szmrecsanyi et al., 2017; Szmrecsanyi and Hinrichs, 2008) and how humans learn these alternations (Bresnan, 2007; Bresnan and Ford, 2010; Campbell and Tomasello, 2001; De Marneffe et al., 2012). For both datives and genitives, the alternation can be predicted with high accuracy through a logistic regression model on hand-labeled features including the animacy, definiteness, givenness, pronominality, and length of noun phrase arguments (Bresnan et al., 2007; Szmrecsanyi and Hinrichs, 2008). While these features are universally important in determining these alternations in English, they are sensitive to the variety of English and the era that it is spoken in (Szmrecsanyi et al., 2017).

The similarity between datives and genitives is evident in terms of both semantics and predictive modeling. In terms of semantics – at least for the instances that are typically included in alternation analyses – both can attribute one nominal argument to another in a possession-type relation: prototypical genitives state such a relation, while datives

often express a change in such a relation (Wolk et al., 2013). This semantic overlap is further evidenced by the fact that the dative and the genitive cases have merged into one in some Indo-European languages such as Greek or Bulgarian (Catasso, 2011; Stolk, 2015). In terms of predictive modeling, both alternations are sensitive to a common set of features, in similar ways, which is reflected in qualitatively similar coefficients for such features in logistic regression models (Szmrecsanyi et al., 2017; Wolk et al., 2013). In both datives and genitives, there are probabilistic tendencies to order short, animate, and/or definite noun phrase arguments before long, inanimate, and/or indefinite ones.

2.2 BERT and syntactic knowledge

BERT (Bidirectional Encoder Representations from Transformers; Devlin et al., 2019) utilizes a bidirectional attention-based architecture to capture dependencies between words. Its design is particularly well suited for capturing relations between words that are linearly distant in a stream of text, which can present issues for traditional sequence-to-sequence (RNN and LSTM) models. Such long-distance relations are invoked in probabilistic accounts of the dative and genitive alternations through the comparison of features across phrasal arguments (e.g., animacy of the recipient and theme), since those features are typically primarily cued by just one word in a phrase that may be arbitrarily long. We expect BERT to possess an understanding of English word order preferences because previous work has shown that they can be learned by structurally-simpler RNN models (Futrell and Levy, 2019).

Past studies have established that BERT’s embeddings encode information about syntactic structure and semantic roles (Jawahar et al., 2019; Manning et al., 2020; Rogers et al., 2021), including at the construction level (Tayyar Madabushi et al., 2020). They also encode information about higher-order organization of the grammatical system that cannot be inferred from any single sentence (Papadimitriou et al., 2021). This information is represented in a multifaceted and gradient manner, much like is posited for human syntactic knowledge, suggesting that insights from human syntactic knowledge may help us understand BERT embeddings and that modeling based on BERT embeddings may help us test hypotheses about human syntactic knowledge.

3 Methods

3.1 Data

Our experiments make use of dative constructions (for training and testing) and genitive constructions (for transfer). To control for effects of variety and era, we restrict our focus to constructions taken from contemporary New Zealand English, as represented by the Canterbury Corpus component of the Origins of New Zealand English corpus (ONZE; Gordon et al., 2007). These constructions occurred in sociolinguistic interviews with New Zealand English speakers born between 1926 and 1987, which were conducted between 1994 and 2007.

Our data consists of 790 datives (680 NP-datives and 110 PP-datives) and 1842 genitives (664 s-genitives and 1178 of-genitives). These are largely the same constructions contained in the data shared by Szmrecsanyi et al. (2017), with minor differences in numbers due to slightly different inclusion criteria. There are two main differences between our data and Szmrecsanyi et al.’s: (1) for the datives, our data is focused on contemporary constructions across a wide range of dative verbs, whereas Szmrecsanyi et al.’s data includes historical constructions and is restricted to datives involving the verb *give*; and (2) for both the datives and genitives, our data contains a brief context for each construction, consisting of the entire line in the corpus from which the construction was extracted, whereas Szmrecsanyi et al.’s data has no context for New Zealand English constructions.

We preprocessed the data by removing transcription annotations that marked pauses, hesitations, and disfluencies. We kept filler words such as ‘um’ and ‘uh’, which are argued to be planned components of an utterance (Clark and Fox Tree, 2002).

3.2 Models

We use two models to predict the relative order of two arguments in a dative construction. Both models consist of a binary classifier that uses pre-trained BERT embeddings as input. The embeddings used by each model represent different syntactic entities and have access to different amounts of context. The *contextless* model uses embeddings that represent the phrasal arguments, each taken in isolation without consideration of the construction or any broader context. The *preference* model uses embeddings that represent different alternants of the entire construction, considered within a broader context. The corresponding formulations of the

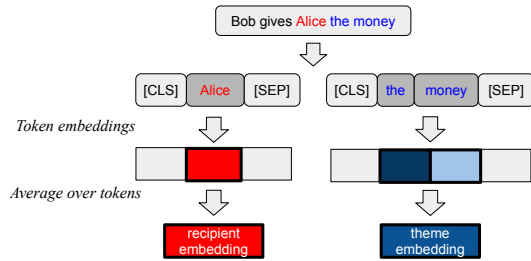


Figure 1: Extraction of embeddings for the contextless model (a)

prediction task undertaken by each model are as follows:

- CONTEXTLESS: predict phrase order from out-of-context phrasal embeddings.** Given the BERT embeddings of the recipient and theme extracted in isolation, i.e. the embeddings of BERT("[CLS] [recipient] [SEP]") or BERT("[CLS] [theme] [SEP]"), determine the order in which the noun phrases appear in a dative construction. See Figure 1 for an illustration of the recipient and theme embeddings.
- PREFERENCE: predict attested alternant from contextual construction embeddings.** Given the BERT embeddings of both alternants of a dative construction extracted in context, i.e. the average of embeddings over the bolded tokens in BERT("[CLS] [context] [verb] [recipient] [theme] [SEP]") and BERT("[CLS] [context] [verb] [theme] to [recipient] [SEP]"), determine which alternant is attested. See Figure 2 for an illustration of the attested and unattested construction embeddings.

The classifier in each model is implemented as a multilayer perceptron with a single hidden layer of size 64 and a sigmoid output layer. For the contextless model, the input is the embedding of the theme concatenated to the embedding of the recipient, and the expected output is 0 if the input is from an NP-dative and 1 if the input is from a PP-dative. For the preference model, the input is the embedding corresponding to the PP-dative concatenated to the embedding corresponding to the NP-dative, and the expected output is 0 if the NP-dative is attested and 1 if the PP-dative is attested.

Each classifier is trained with a binary cross-entropy loss function, via stochastic gradient descent with learning rate 0.01 over 25 epochs. The

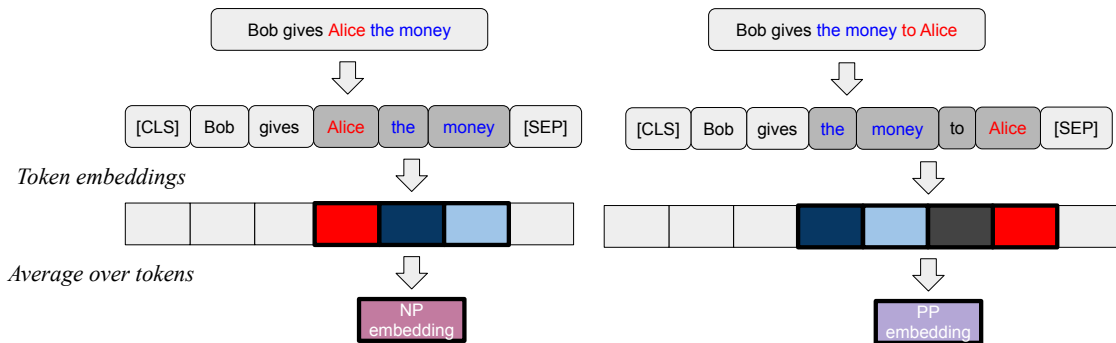


Figure 2: Extraction of embeddings for the preference model (b)

training set for each classifier consists of the same fixed sample of 50 NP-datives and 50 PP-datives, and the held-out test set consists of the same fixed sample of 60 NP-datives and 60 PP-datives. These sizes were chosen to maintain a balance between NP- and PP-datives in training and testing; they are forced to be small by the fact that the data contains only 110 PP-datives. Despite the small size of the training set, we show that the dative alternation can still be reliably predicted without overfitting.

3.3 Embeddings

The embeddings used as input to the models are obtained from the pretrained BERT-base-uncased model. To obtain a single embedding for a phrase or construction, we average the embeddings of all tokens it contains.

For both models, the embeddings are obtained from text sequences that are not single, complete sentences. Since BERT is trained on complete sentences, the embeddings therefore represent unaccounted-for situations and may not be entirely robust. Nevertheless, this situation is unavoidable for various reasons. In the contextless model, embeddings are obtained from phrases, paralleling the use of decontextualized phrases in analyses using hand-labeled features; using complete sentences would introduce context, breaking this parallelism, and would allow the model to ‘cheat’ by referring to information about the relative position of the phrases in position embeddings. In the preference model, embeddings are obtained from lines in the transcripts of a spoken conversational corpus, which may correspond to a fragment of a sentence or several sentences; using complete sentences is not feasible as the transcripts do not indicate sentence boundaries, since utterances in

spontaneous speech are not consistently structured into sentences (e.g., Miller and Weinert, 1998).

The BERT model has a lexical layer (layer 0) and 12 Transformer layers (layers 1–12), meaning that it can produce 13 embeddings for each token, each integrating context to different extents. Our analysis compares the results of using these different embeddings in each model. Thus, we train 26 distinct classifiers in total, corresponding to each of the two prediction tasks (a) and (b), and each BERT layer $l = 0, 1, \dots, 12$.

4 Experiment I: Predicting the dative alternation

In our first experiment, we examine how well the two BERT models are able to predict the dative alternation in the test set. In this examination, we consider the BERT models relative to a logistic regression model based on hand-labeled features, which is the predominant model used to analyze and interpret the alternation in past literature (e.g. Bresnan et al., 2007; Szmrecsanyi et al., 2017). This baseline both establishes how to interpret the performance of the BERT models and highlights the features that are particularly predictive in our training data.

4.1 Baseline logistic model

The baseline logistic regression model is trained on the same balanced training set of 100 dative constructions as the BERT models. Like the contextless BERT model, it receives representations of the recipient and theme as input and must predict the order in which they occur, where the expected output is 0 if the recipient comes first (NP-dative) and 1 if the theme comes first (PP-dative). However, unlike the contextless model, the input repre-

sentations it uses are not machine-learned embeddings but rather vectors of hand-labeled features, derived from variables that have been established as relevant in past work. These variables include definiteness (*indefinite* or definite), pronominality (*nonpronoun* or pronoun), animacy (*inanimate* or animate), and number (*plural* or singular) of both the recipient and the theme, person (*nonlocal* or local) of the recipient, concreteness (*nonconcrete* or concrete) of the theme, and the length difference in orthographic words between the recipient and the theme ($\log \text{recipient.length} - \log \text{theme.length}$).¹

For each categorical variable listed above, the italicized level serves as the reference level; that is, the italicized level has a feature value of 0, while the non-italicized level has a feature value of 1. In each case, the non-reference level is the one that has been argued to be ‘easier’ for lexical retrieval in production planning. Consequently, according to accessibility-based approaches such as Easy First (Bock, 1982; MacDonald, 2013), in which ‘easy’ elements are ordered before ‘hard’ ones, we expect recipient-oriented coefficients to be negative when significant and theme-oriented coefficients to be negative when significant. Similarly, given that shorter phrases are ‘easier’ than longer ones, we expect the length difference coefficient to be positive when significant.

The coefficients learned by the logistic regression model are shown in Table 1. They are qualitatively consistent with results from Bresnan et al. (2007) in terms of both directionality² and significance. There is only one difference, in that recipient definiteness is significant in Bresnan et al.’s results but not in ours; this is likely due to the differences in training data size. This difference notwithstanding, the coefficients are consistent with expectations from Easy First, indicating that Easy First preferences are learnable from our training set.

4.2 Results: model comparison

The baseline logistic regression model achieves an accuracy of 0.86 on the test set. The contextless BERT model achieves a similar accuracy and the preference BERT model far exceeds it, in both cases regardless of the BERT layer that is used to

¹Note that our list of variables differ from that of Bresnan et al. (2007), since we have only included variables pertaining to the recipient and theme and have omitted variables pertaining to the dative verb.

²Note that our coefficients are designed to have the opposite signs to those reported by Bresnan et al. (2007), because we have chosen opposite reference levels.

Table 1: Logistic regression coefficients learned from the training set; bolded coefficients are significant at $p < .05$

	Coeff	z
constant	1.71	
rec.def	-0.20	-0.65
rec.pron	-2.31	-4.75
rec.person	0.31	0.67
rec.anim	-0.67	-2.23
rec.number	0.62	1.55
thm.def	1.00	1.99
thm.pron	0.95	2.32
thm.anim	0.00	0.03
thm.number	-0.15	-0.37
thm.conc	-0.25	-0.50
length diff (log)	1.42	1.69

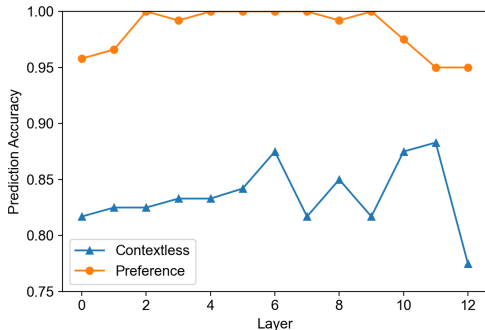


Figure 3: Dative alternation prediction accuracy on the test set by layer

provide input embeddings (Figure 3). In all cases, the accuracies are far above those expected from random chance (0.5), indicating that any overfitting due to the small size of the training set is limited.

At the best BERT layers, the contextless model’s prediction accuracy is 0.88, which exceeds that of the baseline logistic regression model. As the confusion matrices in Table 2 show, the pattern of responses from the contextless model is very similar to the pattern from the baseline model. Thus, the use of contextless BERT embeddings yields classifications that are equivalent to, or better than, the use of hand-labeled features, at a fraction of the annotation cost.

The predictions made by the contextless model are also highly consistent with those made by the

Table 2: Confusion matrices for the logistic model and contextless model on the dative test set

		True Labels		Total
		NP	PP	
Logistic Predictions	NP	53	10	63
	PP	7	50	57
<hr/>				
Contextless Predictions	NP	56	10	66
	PP	4	50	54
Total		60	60	120

logistic model. The models agree on all but 7 constructions in the test set, consisting of 5 NP-datives that are correctly predicted by the contextless model but not by the logistic model and 2 PP-datives that are correctly predicted by the logistic model but not by the contextless model. Thus, the similarity in overall accuracy reflects a similarity in predicting individual alternations, which may imply that the contextless model is self-discovering sensitivities to a similar set of features as the logistic model (i.e., those listed in Table 1).

The preference model does even better than the contextless model, with near-perfect³ accuracy on the test set over several BERT layers. We suspect that this increase in performance of the preference model over the contextless model is due to its incorporation of information about the dative verb and the broader context. Because the accuracy is so high, we do not decompose it further.

5 Experiment II: Zero-shot transfer to genitives

Section 4 showed that the BERT models could successfully predict the dative alternation. In particular, the preference model showed near-perfect classification performance on the test set. Here, we ask whether this best-performing model seems to have learned preferences that are specific to the dative alternation or more general preferences that also apply to the genitive alternation.

5.1 The transfer setup

To enact transfer, we created input embeddings for the genitive data in the same way as for the dative data Section 3.3, under the alignment of s-genitives with NP-datives and of-genitives with PP-datives.

³We do not interpret accuracies of 1 as ‘perfect’ due to the limited sample size of the test set. In a larger and more diverse test set, we expect the preference model’s accuracy to be high but not quite this extreme.

Table 3: Confusion matrices for the adjusted outputs of the preference model on the genitive dataset

		True Labels		Total
		S	Of	
Preference Predictions	S	489	312	801
	Of	175	866	1041
Total		664	1178	1842

That is, for each attested genitive in our dataset, we manually created its unattested alternant and obtained embeddings for both the attested and unattested alternants in context. We then formed the input to the preference model by concatenating the embedding corresponding to the of-genitive to the embedding corresponding to the s-genitive.

We measure the success of the transfer by how well the classifier separates the s- and of-genitive constructions. To do so, we manually move the decision threshold by applying an additional linear translation before the final sigmoid layer. We pick the threshold value that yields equal accuracy for s- and of-genitives and treat the overall accuracy obtained under this threshold as our measure of success.

5.2 Results: transfer accuracy

The preference model trained on layer 2 of BERT achieves the best adjusted transfer accuracy of 0.74, which is significantly better than the baseline accuracy of 0.64 achieved by only predicting of-genitives ($p < 0.001$ by exact binomial test). The confusion matrix of the transfer is shown in Table 3, and a graph of its prediction outputs over the entire genitive dataset is shown in Figure 4. While the model is able to separate s- and of-genitives fairly well, suggesting that it has learned general ordering constraints from datives that are applicable to genitives, its output probabilities are compressed, suggesting that these general constraints may yield only weak preferences that could be further adapted for specific constructions.

5.3 Association between labels and features

To dig into the general constraints underpinning the transfer performance, we now consider how the preference model is influenced by the features that have been recognized as (potentially) relevant for predicting both dative and genitive alternations. These target features are animacy and definiteness of the possessor (recipient), animacy of the possesum (theme), and difference in argument lengths

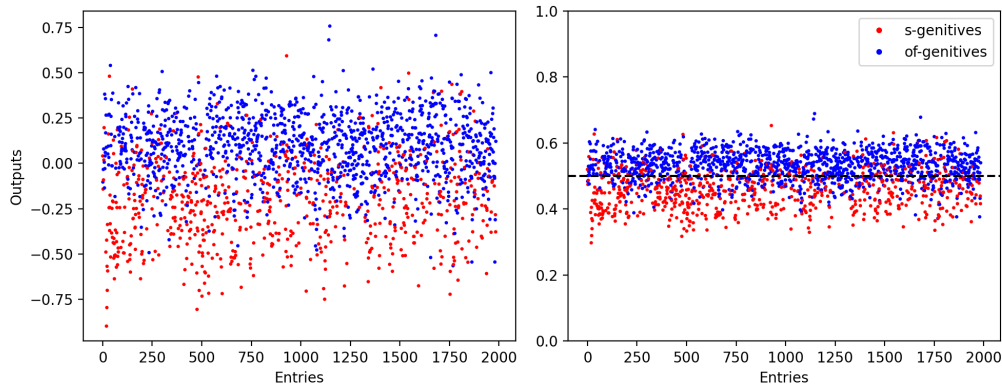


Figure 4: Genitive alternation predictions by the preference model. Left: pre-sigmoid outputs; Right: sigmoid adjusted outputs

(possessor – possessum)⁴.

For each target feature, we restrict attention to a subset of constructions differing only in that feature, to minimize confounds. The other features are fixed at levels that maximize the size of this subset and ensure that each level of the target feature is (maximally) attested. Then, for each level of the target feature, we calculate its pointwise mutual information (PMI) with both the BERT labels predicted by the preference model and true labels of the chosen alternant in each construction in the subset. If the associations are consistent with the principle of Easy First, we expect animate and definite possessors, inanimate possessums, and small/negative length differences to yield positive PMI with s-genitives and negative PMI with of-genitives, and vice versa for the opposite levels of each feature. If the preference model has learned associations that are present in genitives, despite being trained on datives, then we expect the PMIs with the BERT labels to pattern similarly to the PMIs with the true labels.

The results are shown in Tables 4 to 7. The associations between features and genitive alternant choice do not consistently align with expectations from Easy First, either for the alternant labels predicted by the preference model or for the true labels. It is hard to know whether this unexpected behavior indicates a real quirk of New Zealand English or is just an artifact of the sparse data and/or the specific

levels at which non-target features were fixed.

Regardless, the PMIs with the model’s labels almost always agree in sign and relative magnitude with the PMIs with the true labels, which suggests that the model has learned general associations that are transferable between the dative and genitive alternations. The associations seem to be weaker for the model than for the true labels, consistent with the idea that general constraints on order preferences are weaker than construction-specific constraints. However, the associations with possessor definiteness (Table 6) appear to be stronger for the model than for the true labels, which is especially surprising given that recipient definiteness was not strongly correlated with alternant choice in the datives training data (Table 1).

⁴In order to permit alternation, genitive constructions must have a definite possessum (Rosenbach, 2014). This definiteness can be marked by determiner in of-genitives, but not in s-genitives (e.g., [the tires] of the car vs. the car’s [tires]). To account for this difference when calculating length, we followed past work (e.g., Szmrecsanyi and Hinrichs, 2008) in not counting *the* at the beginning of the possessum in of-genitives.

Table 4: Transfer accuracy and PMIs on genitive constructions differing only in possessor animacy. These constructions all have definite possessors, inanimate possessums, and a length difference of 1.

Possessor animacy			
s-genitive	Inanim	Anim	Total
# Correct	8	78	86
# Total	15	100	115
Accuracy	0.53	0.78	0.75

of-genitive	Inanim	Anim	Total
# Correct	379	23	402
# Total	457	28	485
Accuracy	0.83	0.82	0.83

BERT-labels PMI	Inanim	Anim
s-genitive	-0.63	1.20
of-genitive	0.19	-1.03

True-labels PMI	Inanim	Anim
s-genitive	-2.59	2.03
of-genitive	0.26	-1.89

Table 5: Transfer accuracy and PMIs on genitive constructions differing only in possessum animacy. These constructions all have animate and definite possessors and a length difference of 1.

Possessum animacy			
s-genitive	Inanim	Anim	Total
# Correct	78	35	113
# Total	100	46	146
Accuracy	0.78	0.76	0.77

of-genitive	Inanim	Anim	Total
# Correct	23	1	24
# Total	28	2	30
Accuracy	0.82	0.50	0.80

BERT-labels PMI	Inanim	Anim
s-genitive	-0.06	0.15
of-genitive	0.12	-0.37

True-labels PMI	Inanim	Anim
s-genitive	-0.09	0.21
of-genitive	0.36	-2.03

Table 6: Transfer accuracy and PMIs on genitive constructions differing only in possessor definiteness. These constructions all have animate possessors, inanimate possessums, and a length difference of 1.

Possessor definiteness				
s-genitive	Indef	Def	Def-pn	Total
# Correct	30	78	5	113
# Total	35	100	10	145
Accuracy	0.86	0.78	0.50	0.78

of-genitive	Indef	Def	Def-pn	Total
# Correct	4	23	2	29
# Total	8	28	3	39
Accuracy	0.50	0.82	0.67	0.74

BERT-labels PMI	Indef	Def	Def-pn
s-genitive	0.24	-0.04	-0.53
of-genitive	-0.66	0.08	0.70

True-labels PMI	Indef	Def	Def-pn
s-genitive	0.05	-0.01	-0.03
of-genitive	-0.19	0.05	0.12

Table 7: Transfer accuracy and PMIs on genitive constructions differing only in length difference (possessor – possessum). These constructions all have inanimate and definite possessors and inanimate possessums.

Length difference				
s-genitive	≤ 0	$= 1$	≥ 2	Total
# Correct	3	8	3	14
# Total	11	15	5	31
Accuracy	0.27	0.53	0.60	0.45

of-genitive	≤ 0	$= 1$	≥ 2	Total
# Correct	88	379	90	557
# Total	130	457	127	714
Accuracy	0.68	0.83	0.71	0.78

BERT-labels PMI	≤ 0	$= 1$	≥ 2
s-genitive	0.48	-0.33	0.40
of-genitive	-0.18	0.09	-0.14

True-labels PMI	≤ 0	$= 1$	≥ 2
s-genitive	0.91	-0.39	-0.14
of-genitive	-0.06	0.01	0.01

6 Discussion & Conclusion

In this paper, we have presented two models designed to predict dative alternations from BERT embeddings. In Section 4, we found that the dative alternation can be predicted with high accuracy from BERT embeddings, and in a manner mostly consistent with traditional logistic regression models based on hand-annotated features. In Section 5, we explored the zero-shot transferability of our context-aware dative alternation model to genitive alternations. The transfer was relatively successful, and we explored both its success and limitations by analyzing the pointwise mutual information between assigned labels and features. Our findings suggest that BERT-based alternation models perform comparably to traditional approaches utilizing hand-annotated features, and that they are capable of recognizing general principles that yield similarities between the dative and genitive alternations.

Our experiments showcase potential approaches for understanding how word-order preferences are encoded in BERT’s embedding space and the extent to which they are construction-specific. The success of our preference model in the zero-shot transfer from datives to genitives suggests that it is not solely relying on (dative) construction-specific constraints to derive word-order preferences, but rather appealing to more general constraints. One possible such general constraint is Easy First (Bock, 1982; MacDonald, 2013), which showed reasonable explanation of patterns of alternant choice in our datives training set. However, the fact that the transferred model captures the apparent patterns in genitive alternant choices even when they do not seem to be consistent with Easy First suggests that the general constraints it learned from the datives cannot be boiled down just to Easy First. Given that the preference model utilizes pre-trained embeddings of entire alternants, which plausibly reflect in some way the extent to which lexical subsequences within that alternant are evidenced in BERT’s training data, it is possible that the model’s choices may be influenced by local surprisal statistics based on the different lexical subsequences that are formed when the noun phrase arguments are placed in different orders. That is, the general constraints being invoked may involve some degree of ‘episodic memory-matching’ based on BERT’s pre-training data, as well as consideration of more abstract features.

One interesting future study could consider a

direct comparison between the alternation preferences of the preference model with that of humans. In the present work, we focused on analyzing the extent to which our BERT-based models can determine the order in which humans produce two noun phrases in dative and genitive constructions. To what extent does learning to match these categorical production preferences enable the prediction of gradient human perceptual preferences? Humans have preferences about reading the arguments in one order relative to the other, which varies between individuals and across contexts (Bresnan and Ford, 2010). By evaluating the similarities and differences between these preferences and the probabilities output by the preference model, we may be able to further understand both BERT embeddings and human syntactic knowledge.

7 Limitations

Although a small training set of 100 dative constructions appears to be sufficient for predicting dative alternations and for zero-shot transfer to genitive alternations, we ideally want a larger training set to improve the robustness of our models. Also, due to the strong correlation between animacy and alternation type in both the dative and genitive datasets, obtaining a sufficient number of constructions that differ minimally in features for the PMI analysis is challenging. Some of the feature labeling in our dataset may also be too coarse to capture the gradient nature of the features. For instance, rather than treating animacy to be binary, Szmrecsanyi et al. (2017) considers human and animals, collective, temporal, locative, and inanimate as distinct categories. All of these data-related issues can add variability to our analysis.

On the model side, our interpretation of results has generally made the assumption that our models are actually making predictions from self-discovered versions of the features that the literature has shown to be relevant to the dative and genitive alternations, rather than from something else entirely. Although our models’ predictions are consistent with known associations between features and alternations, it does not necessarily imply that they are learning to be sensitive to those features, since the training labels are themselves correlated with the features. In addition, we have interpreted our results very generally, but the restriction to contemporary New Zealand English may limit the generalizability of our findings.

Acknowledgements

Cloud computing credits for this study were provided through a UCSB Undergraduate Research & Creative Activities grant to Qing Yao. The data used in this study was collected by Simon Todd with support from the National Science Foundation under Grant No. BCS-1025602, with the help of Kirsty Canillo and Daniel Bürkle and the guidance of Joan Bresnan, Anette Rosenbach, and Jen Hay. We gratefully acknowledge the students and researchers from the Linguistics Department of the University of Canterbury who contributed to the creation of the contemporary Canterbury Corpus portion of the ONZE Corpus, as well as Robert Fromont for programming and maintaining the LaBB-CAT interface to the corpus that was used to extract our data. We thank Marthe Midtgaard and Shialan Yu for their contributions to early work from which this study developed, Mary Bucholtz and members of the UCSB Linguistics undergraduate thesis seminar for their support, and members of the UCSB CEILING research group for their feedback on an early draft.

References

- J. Kathryn Bock. 1982. [Toward a cognitive psychology of syntax: Information processing contributions to sentence formulation](#). *Psychological Review*, 89(1):1–47.
- Joan Bresnan. 2007. [Is syntactic knowledge probabilistic? Experiments with the English dative alternation](#). In Sam Featherston and Wolfgang Sternefeld, editors, *Roots: Linguistics in Search of its Evidential Base*, pages 77–96. Mouton de Gruyter, Berlin.
- Joan Bresnan, Anna Cueni, Tatiana Nikitina, and R. Harald Baayen. 2007. [Predicting the dative alternation](#). In Gerlof Bouma, Irene Krämer, and Joost Zwarts, editors, *Cognitive Foundations of Interpretation*, pages 69–94. KNAW, Amsterdam.
- Joan Bresnan and Marilyn Ford. 2010. [Predicting syntax: Processing dative constructions in American and Australian varieties of English](#). *Language*, 86(1):168–213.
- Aimee L. Campbell and Michael Tomasello. 2001. [The acquisition of English dative constructions](#). *Applied Psycholinguistics*, 22(2):253–267.
- Nicholas Catasso. 2011. [Genitive-dative syncretism in the Balkan sprachbund: An invitation to discussion](#). *SKASE Journal of Theoretical Linguistics*, 8(2):70–93.
- Herbert H. Clark and Jean E. Fox Tree. 2002. [Using uh and um in spontaneous speaking](#). *Cognition*, 84(1):73–111.
- Marie-Catherine De Marneffe, Scott Grimm, Inbal Arnon, Susannah Kirby, and Joan Bresnan. 2012. [A statistical model of the grammatical choices in child production of dative sentences](#). *Language and Cognitive Processes*, 27(1):25–61.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of NAACL-HLT 2019*, pages 4171—4186. Association for Computational Linguistics.
- Holger Diessel. 2020. [A dynamic network approach to the study of syntax](#). *Frontiers in Psychology*, 11:604853.
- Richard Futrell and Roger P. Levy. 2019. [Do RNNs learn human-like abstract word order preferences?](#) *Proceedings of the Society for Computation in Linguistics*, 2:50–59.
- Elizabeth Gordon, Margaret MacLagan, and Jennifer Hay. 2007. [The onze corpus](#). In Joan C. Beal, Karen P. Corrigan, and Hermann L. Moisl, editors, *Creating and Digitizing Language Corpora: Volume 2: Diachronic Databases*, pages 82–104. Palgrave Macmillan, Basingstoke.
- Stefan Th. Gries and Anatol Stefanowitsch. 2004. [Extending collocation analysis: A corpus-based perspective on ‘alternations’](#). *International Journal of Corpus Linguistics*, 9(1):97–129.
- Jess Gropen, Steven Pinker, Michelle Hollander, Richard Goldberg, and Ronald Wilson. 1989. [The learnability and acquisition of the dative alternation in English](#). *Language*, 65(2):203–257.
- Ganesh Jawahar, Benoît Sagot, and Djamel 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. Association for Computational Linguistics.
- Maryellen C. MacDonald. 2013. [How language production shapes language form and comprehension](#). *Frontiers in Psychology*, 4:226.
- Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. [Emergent linguistic structure in artificial neural networks trained by self-supervision](#). *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.
- Jim Miller and Regina Weinert. 1998. *Spontaneous Spoken Language: Syntax and Discourse*. Oxford University Press, Oxford.

- Isabel Papadimitriou, Ethan A. Chi, Richard Futrell, and Kyle Mahowald. 2021. [Deep subjecthood: Higher-order grammatical features in multilingual BERT](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2522–2532. Association for Computational Linguistics.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. [A primer in BERTology: What we know about how BERT works](#). *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Anette Rosenbach. 2014. [English genitive variation — the state of the art](#). *English Language and Linguistics*, 18(2):215–262.
- Joanne Vera Stolk. 2015. [Dative by genitive replacement in the Greek language of the papyri: A diachronic account of case semantics](#). *Journal of Greek Linguistics*, 15(1):91–121.
- Benedikt Szmrecsanyi, Jason Grafmiller, Joan Bresnan, Anette Rosenbach, Sali Tagliamonte, and Simon Todd. 2017. [Spoken syntax in a comparative perspective: The dative and genitive alternation in varieties of English](#). *Glossa: a journal of general linguistics*, 2(1).
- Benedikt Szmrecsanyi and Lars Hinrichs. 2008. [Probabilistic determinants of genitive variation in spoken and written English](#). In Terttu Nevalainen, Irma Taavitsainen, Päivi Pahta, and Minna Korhonen, editors, *The Dynamics of Linguistic Variation: Corpus Evidence on English Past and Present*, pages 291–309. John Benjamins, Amsterdam.
- Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. [CxGBERT: BERT meets Construction Grammar](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032. International Committee on Computational Linguistics.
- Daphne Theijssen, Louis Ten Bosch, Lou Boves, Bert Cranen, and Hans Van Halteren. 2013. [Choosing alternatives: Using Bayesian networks and memory-based learning to study the dative alternation](#). *Corpus Linguistics and Linguistic Theory*, 9(2):227–262.
- Christoph Wolk, Joan Bresnan, Anette Rosenbach, and Benedikt Szmrecsanyi. 2013. [Dative and genitive variability in Late Modern English: Exploring cross-constructional variation and change](#). *Diachronica*, 30(3):382–419.

Language Models and the Paradigmatic Axis

Timothee Mickus

University of Helsinki

timothee.mickus@helsinki.fi

Abstract

The massive relevance of large language models, static, and contextualized word embeddings in today’s research in NLP implies a need for accounts of how they process data from the point of view of the linguist. The goal of the present article is to frame language modeling objectives in structuralist terms: Word embeddings are derived from models attempting to quantify the probability of lexical items in a given context, and thus can be understood as models of the paradigmatic axis. This reframing further allows us to demonstrate that, with some consideration given to how to formulate a word’s context, training a simple model with a masked language modeling objective can yield paradigms that are both accurate and coherent from a theoretical linguistic perspective.

1 Introduction

It is a truism to say that field of natural language processing (NLP) has seen profound changes over the past decade. The development of static neural word embeddings, the introduction of contextualized embeddings, and their re-branding as large language models are as many steps along this transition, and each have yielded many impressive technical advancements over the prior state of the art.

It is also a truism to say that this technical progress stems for the most part from an engineering culture, and that the concerns stressed as more prominent in NLP have primarily to do with the maturing technology of deep learning—much of the ongoing background discussion in NLP centers on questions such as scaling up (Sutton, 2019), or defining tasks to solve and metrics to optimize (Tedeschi et al., 2023; Ganesh et al., 2023). The current concerns of NLP pertain not to language, but to what can be achieved through language.

At the same time, there is a sizable body of work interested in discovering what aspects of language

are encoded in language models and word embeddings alike. Many adopt as their main angle of research treating language models as or comparing them to language speakers (e.g., Linzen et al., 2016)—to identify whether they encode some specific linguistic information (e.g., Hewitt and Manning, 2019; Chi et al., 2020); contrast these models with what actual speakers do (Bender and Koller, 2020); or characterize what they can and cannot capture (Merrill et al., 2022; Bouyamourn, 2023).

Underlying all this work on evaluating NLP models is the linguistic framework they are instances of—namely, distributional semantics. Tackling this subjects are technical accounts and surveys (a.o., Lenci, 2018), pieces discussing their usefulness to theoretical linguistics (e.g., Boleda, 2020), works underscoring the theoretical limitations of distributional models (e.g., Emerson, 2020), historical reviews of how this framework has evolved (Brunila and LaViolette, 2022). Yet, conceptual discussions of the distributional framework itself are surprisingly hard to find: Proposed extensions of distributional semantics more often than not focus on incorporating extraneous elements from more strictly formalized frameworks (e.g., Baroni et al., 2014; McNally, 2017; Herbelot and Copestake, 2021), rather than conceptualizing and formalizing distributional methods in and of themselves. This fact is all the more surprising once we factor in that the impressive successes of modern language models are achieved through purely distributional means.

In this paper, we build upon Sahlgren (2008), Gastaldi (2021) and Gastaldi and Pellissier (2021), who keenly analyzes the links between word embeddings, distributional semantics and structuralism. We argue here that systems trained on language modeling objectives can be understood in structuralist terms as *models of the paradigmatic axis*. Sahlgren, Gastaldi and Gastaldi and Pellissier also stress the link between structuralism and distributionalism. Unlike Sahlgren and Gastaldi, we do

not conflate distributional models with vector space semantics; and whereas [Gastaldi and Pellissier](#) connect paradigms and word embeddings through a reformalization of the concept of paradigm with the explicit goal of deriving structural representations, we argue that there is an obvious and immediate link between the language modeling objective and a paradigmatic axis and that this relationship can be attested empirically.

We first include a short historical account of distributionalism and a more substantiated description of our suggested framework for embeddings and language models in Section 2. We then provide empirical demonstrations of how basic linguistic considerations can shape the properties observed in language models in Section 3.

2 Language models and paradigms

We first start by gathering here some key elements of structuralist theory to provide the reader with all the relevant context; more thorough accounts can be found in [Brunila and LaViolette \(2022\)](#), [Sahlgren \(2008\)](#) and [Gastaldi \(2021\)](#).

Structuralism and the paradigmatic dimension of language. The birth of structuralism in linguistic is usually attributed to [Saussure \(1916\)](#). One chief concern underpinning it is the study of language for language’s sake ([Gastaldi, 2021](#)), which it achieves by making its central object of study the *structure* of the language. In short, the structuralist program, as framed by [Saussure \(1916\)](#), involves the following tenets: (i) that a language has a structure relating sound and meaning; (ii) that this structure can be established by isolating the signs of this language; and (iii) that to isolate signs, one needs to show that variation in sound (or meaning) entails variation in meaning (or sound).

Signs can be related to one another in a variety of ways; one we are especially vested in is that of a paradigmatic relation, as formalized by [Hjelmslev \(1971\)](#). Simply put, words that compete for the same position in a context are said to form a paradigm. Consider for instance ex. (1):

(1) I am teaching.

Notice how the word ‘teaching’ could have been replaced by some other word not attested in ex. (1), be it ‘writing’, ‘dancing’ or ‘fabulous.’ The relationship between ‘teaching’ and these other candidate words is one “*in absentia*,” that is, between

terms as members of the sign inventory of a language, rather than between terms co-occurring in a context. This contrasts with relationships that hold between terms in the same context, usually referred to as “syntagmatic”—consider for instance how in ex. (1) the word ‘I’ is necessary because of how it relates to the word ‘am,’ that is to say, this relationship holds “*in praesentia*.”

The notion of paradigm found in [Hjelmslev \(1971\)](#) builds upon [Saussure’s \(1916\)](#) conception of associative series: [Saussure](#) highlights that we can associate series of words based on whether they share common formal elements (‘teaching’, ‘teaches’, ‘teacher’, ‘teach’, ...), have similar meanings (‘teaching’, ‘learning’, ‘education’, ...), or display formal similarities by happenstance (‘teach’, ‘peach’, ‘beach’, ...). As noted by [van Marle \(1984\)](#), this entails that [Saussure’s \(1916\)](#) view is “that the paradigmatic dimension of language is simply highly indefinite and undetermined” (p. 12). The position we defend here is that a fruitful application of the structuralist concept of paradigms or series to modern NLP only requires a Hjelmslevian take on paradigms. In practice, we will consider a paradigm to be a relationship in *absentia* between terms that are equally syntagmatically constrained.

Distributionalism. Distributionalism is a specific strand of American structuralism best exemplified by the figures of Bloomfield and Harris. Their main contribution to structuralism is a deeper focus on what the study of co-occurrences of items (be they signs, words, morphemes or phonemes) and their distributional regularities can highlight.

Harris, in particular, had a keen interest in formalizing linguistics as an empirical, objective science, for which he deemed imperative that observations be carried out as methodically as possible ([Léon, 2011](#)). A seminal example was provided in [Harris \(1954\)](#), where he argued that the analysis of co-occurrences of linguistic elements suffices to establish a structural description of a language.

One notion of interest in [Harris’s](#) work is that of *distributionally substitutable* elements: It consists in the iterative and methodological construction of sets of predictably interchangeable words. To take a concrete example, consider the context:

(2) On _____, the office is open from 9AM through 5PM.

Across a large corpus analysis, we expect that we

might attest several possible nouns referring to days of the week in the position left blank in ex. (2)—but nothing else. If, across all contexts we encounter them, these words are in fact substitutable, we can group them into a substitution set. This process can be iterated: For instance, if we have already established that days of the week form a substitution set, we can consider examples such as

- (3) The university is closed this Wednesday.
 (4) The library is closed this Sunday.

Here, the contexts of the terms (underlined) can be equated as their differences only involve variation within a substitution set; which would therefore allow us to group the terms ‘university’ and ‘library’ in another substitution set. Remark that elements in a substitution set correspond to different paradigmatic choices (Sahlgren, 2008): In other words, distributional substitutability is an operationalization of the concept of paradigmatic relationships based on the distributions of words in context.

Vector space semantics and distributional semantics models. One early key success of the distributionalist approach was the discovery that distributional similarity correlates well with word similarity judgments (Rubenstein and Goodenough, 1965). This is often referred to as the *distributional hypothesis*: similar words will occur in similar contexts.¹ This novel perspective eventually gave rise to *distributional semantics*, the field studying how (word) distribution differences correlates with (word) meaning differences. However, to make good of this insight, one hurdle to overcome was the computational challenges entailed by a distributional analysis of an entire corpus. The advent of vector-based means of representing linguistic items (Salton et al., 1975; Landauer and Dumais, 1997)

¹Harris himself was fundamentally invested in not relying on meaning and speaker cognition in linguistics (Brunila and LaViolette, 2022), and conceived distributional as strictly distinct from (though correlated with) meaning. This sheds an interesting light on literature surrounding the cognitive plausibility of distributional accounts of language (Miller and Charles, 1991; Landauer and Dumais, 1997; Mandera et al., 2017). Harris’s position is fundamentally at odds with many of the more successful and better studied linguistic frameworks: In particular, Chomsky (1965) frames linguistic as a branch of psychology, which has to be understood as a departure from distributionalism and structuralism. In that respect, approaches attempting to reconcile generativism and distributionalism (e.g., Baroni et al., 2014; Herbelot and Copestake, 2021), have to be put in the light of the distributional semantics enterprise, and have to be understood as departures from the purely distributional approach of Harris (1954).

provided the means necessary to carry out distributional analyses at this scale. As a result, modern expositions of distributional semantics often conflate vector space semantics and distributional models (e.g., Lenci, 2018; Boleda, 2020; though not always, e.g., Erk, 2012). The relation between vector representations and distributional analyses is, however, of a contingent nature—while the usefulness of high-dimensional space for semantic representations was established early on in computationally oriented research communities (Salton et al., 1975; Schütze, 1992), this need not be the sole means by which a distributional analysis can be carried out.

The language modeling objective(s). If vector space models and distributional models should not be conflated, why then should the current spate of embedding and language models be construed as distributional models? A number of the neural models that are discussed in NLP—and in particular most embedding and language models—are derived from word–context co-occurrences. In practice, they try to quantify the probability of a term given its context, or formally:

$$p(t|c) \tag{1}$$

where t corresponds to a target *term*, and c stands for a *context*. What constitutes a term and a context can in principle vary quite a lot: Contexts have been defined by means of sentences, documents, paragraphs, or syntactic trees; whereas terms have been defined either as word, or increasingly commonly as word-pieces, and may or may not factor in spelling information.

Models that do not directly capture the above often instead compute a related quantity, or an information-theoretic variant thereof. For instance, while the CBOW objective of Mikolov et al. (2013) is explicitly eq. (1), the counterpart skip-gram architecture instead models $p(c|t)$; moreover, in practice, the exact objectives used to trained word2vec, the negative sampling and hierarchical softmax objectives, differ from eq. (1). Note however that the former is simply a reformulation of the probability definition, whereas the latter has already been the subject of much analysis, starting with Levy and Goldberg (2014) who related it to PMI-based models. Looking at more recent works, it is also straightforward to identify the masked language modeling introduced by Devlin et al. (2019) as an instance of eq. (1); it also corresponds to the sentinel-based objective of T5 architectures (Raffel

et al., 2020); whereas the ELECTRA architecture of (Clark et al., 2020) is explicitly linked to the negative sampling objective. As for causal language models, it can be identified as a formulation of the usual autoregressive objective $p(w_i|w_{<i})$.

In short, many neural and non-neural NLP systems, as they can be construed as word generators conditioned on other text, fall within the scope of eq. (1). That similar objectives have been used to develop the most prominent tools across the last decade, from static word embeddings to language models,² appears an obvious consequence of the very limited amount of annotations necessary to set up this objective: The sole requirement is that terms be identified within their context—i.e., that the corpus be presegmented in linguistic units.

A definition of distributional models. In what follows, we consider a distributional model to be any system that satisfies the following criteria:

- (i) given a context, it produces a distribution of terms, following eq. (1);
- (ii) this distribution is derived from corpus data;
- (iii) this distribution is applicable beyond the corpus data it was derived from.³

One could consider, as a fourth criterion, requiring that the context does not contain the term—out of concern that the probability $p(t|c)$ would degenerate to assigning 1 to the attested term t and 0 to all other terms. Such a case can only occur if the context is itself segmented (or segmentable) in linguistic units. Document models (e.g., Salton et al., 1975; Landauer and Dumais, 1997) would be ruled out by this fourth criterion.

Distributional models are models of the paradigmatic axis. This can be established by considering the following three facts.

First, that the language modeling objective is fundamentally ambiguous: While it is reasonable to expect that a well-formed model of eq. (1) tends

²One family of models conspicuously absent are those trained with human feedback, such as ChatGPT.

³This third criterion might seem somewhat trivial, but it both reflects the actual practices of the community that builds said models (assessing generalization capabilities on held-out data is a central tenet of the NLP methodology), and constitutes a departure from strict corpus-based accounts of distributional semantics, including Harris (1954) as well as more recent developments. For instance, Baroni et al. (2014) state (p. 247) that “the meaning of content words lies in their distributions over large spans of texts.”

to assign greater probabilities to the terms that are indeed attested in their respective contexts, this expectation is however defeasible, since speakers may elect to use terms that are less common or surprising. Consequently, a model will assign non-zero probability scores to words other than the actual attested term: If we were to provide ex. (2) to a language model, we would not expect it to assign all its mass to a single term (say “Tuesday”) as some other terms could also fit this context (unless we are faced with an acute case of overfitting).

Second, that the model’s learned distribution should be syntagmatically (and semantically) constrained. If we assume our distributional model assigns probabilities in a manner that reflects what humans are likely to produce, then, while we might expect some fundamental ambiguity between possible terms, this ambiguity is not absolute. Going back to what a model would do of ex. (2), we can strongly conjecture that its probability mass would indeed be accumulated on a narrow class of terms, including mostly days of the weeks. Words belonging in this class will necessarily share a number of semantic traits—since by construction all of them are equally adequate in this context, they also have to be semantically compatible with it: In short, the relationship between terms described by the contextual distribution in eq. (1) should in principle capture some aspect of their semantics, as per the distributional hypothesis. We can also point out that the distribution for this context ought to characterize determiners as much more unlikely than nouns, i.e., this contextual constraint is not just semantic in nature, but rather syntagmatic.

Third, that the learned distribution is a relationship in absentia. Which actual term t is attested in a given context c is in fact somewhat irrelevant, as we are dealing a distribution over ambiguous terms. The relation between the output probability distribution and the attested word is thus only a loose indicator of our model’s validity. What we really expect of a language model is that it properly encodes the underlying ambiguity of possible terms in a manner that is coherent with the syntagmatic constraints of the context. As a consequence, the probability distribution therefore encodes a relationship between abstract terms that compete for a given position, and not the relation between the one attested term and its context.

In short, the objective of eq. (1) entails (i) associating a series of ambiguous terms (ii) with similar semantics constrained by the syntagmatic relation-

ships encoded in the context (iii) as a relationship in absentia. Thus, the output probability distribution of a language model describes a relationship between words that is conceptually similar to Saussure’s (1916) associative series, Hjelmslev’s (1971) paradigms and Harris’s (1954) distributionally substitutable elements—or more simply put, distributional models are models of the paradigmatic axis.⁴

Connections with prior works. That word embedding models are related to the structuralist concept of a paradigmatic axis is not an entirely novel idea: Sahlgren (2008) already identified that some (non-neural) word embedding models, especially those which define contexts as windows of words around the target term, instantiate paradigmatic relations. A very similar connection between distributional models and paradigms was also established by Gastaldi and Pellissier (2021), but they do not equate the model’s objective with the structuralist concept. Instead, Gastaldi and Pellissier identify paradigms as a supplementary construct to explain why specific terms co-occur across varied contexts. Their notion of paradigms departs from the usual structuralist concept in two ways: (i) they propose to formalize paradigms by means of syntactic, informational and characteristic content; and (ii) they explicitly formulate paradigms as sets (rather than terms that may be more or less directly associated) that can exhibit some form of hierarchical subclass structure. These theoretical additions are more than justified when considering what they yield: some means of deriving a linguistic structure from pure distributional analysis. However, they also obfuscate the relationship between language modeling objectives and paradigms, which limits the applicability of their conception of paradigmatic relation as an analytical tool for modern NLP systems.

It is worth stressing that the objective eq. (1) also entails some differences with respect to the traditional notion of a paradigm. In particular, the inclusion of a term in an associative series is quantified by the probability assigned to it through eq. (1). While this is in line with the “highly indefinite and underdetermined” view of Saussure (1916), this also starkly contrasts with later developments of this concept—chief of which Harris’s (1954)—

⁴It is tempting to include syntagmatic relations in what distributional models describe (e.g., Sahlgren, 2008). Yet syntagmatic relations are expected to hold between words in the context, given as input. A more appropriate characterization would be that they constrain paradigmatic series: Syntagmatic relations are implicitly captured to explicitly model paradigms.

where for any term we may say whether or not it is part of a paradigm. Distributional models, in contrast, construe the relevance of a term to a specific paradigm as a matter of fuzzy set membership: Some terms are more likely members than others.

3 Empirical confirmation

While the notion that systems designed to satisfy the language modeling objective are models of the paradigmatic axis is an appealing one, we still require some empirical confirmation of its validity.

Our approach will be as follow: train neural networks with a language modeling objective; and then verify whether their output distributions over terms describe reasonable paradigms. To showcase whether this re-framing of language models as models of the paradigmatic axis can be helpful to the linguist, we can also discuss whether manipulating what linguistic information is provided as context modifies performances in a theoretically coherent way. In practice, our focus will be on *positional* information: This has been one of the features separating static embedding models such as word2vec from contextual embedding models such as BERT, and we can strongly expect that models where context is captured as a bag-of-words yield much less accurate representations of the paradigmatic axis than models that properly factor word order. Very relevant prior work by Sinha et al. (2021) already found this positional information to be necessary for high downstream performances.

A direct comparison of off-the-shelf static and contextual embedding models is somewhat meaningless to our particular endeavor, since they vary on many aspects—including but not limited to the data they have been trained on, the number of parameters they contain and the complexity of the computations they perform. As such, we will start by describing in Section 3.1 two closely related architectures for position-aware and position-agnostic language models which we will then train on the same data, so as to provide a meaningful comparison of their outputs in Sections 3.2 and 3.3.

3.1 Architectures

To facilitate our empirical investigation of whether language modeling objectives lead to models of the paradigmatic axis, let us lay out a few design requirements as to how our language models should be conceived. First, to simplify any judgments on the resulting distributions over terms, it is prefer-

able to study models trained on data pre-segmented in words, rather than word-pieces or other types of linguistic units. Second, it is preferable to keep the model conceptually simple so that its computations remain interpretable, although it is also necessary to ensure that the model is expressive enough to produce non-trivial representations of the paradigmatic axis. Third, the model needs to be lightweight enough to guarantee the replicability of our experiments. Fourth and last, as we focus on positional information, we should make sure that ablating all position information does not require a massive overhaul of the network.

Factoring in all these design requirements, we propose two architectures loosely inspired from the Transformer architecture (Vaswani et al., 2017), one *position-agnostic* and the other *position-aware*. In both cases we consider words as terms, contexts are defined as all other words in a sentence (i.e., we consider some form of masked language modeling). Formally, our position-agnostic network can be described as:

$$p(t_i|c, \theta) = \text{softmax} \left(\mathbf{W}^{(\text{proj})} \mathbf{o} \right) \quad (2)$$

$$\mathbf{o} = \phi \left(\mathbf{W}^{(\text{out})} \phi(\mathbf{h}) \right) \quad (3)$$

$$\mathbf{h} = \text{softmax} \left(\frac{\mathbf{q} \cdot \mathbf{K}^T}{\sqrt{d}} \right) \mathbf{V} \quad (4)$$

$$\mathbf{q} = \text{LayerNorm} \left(\mathbf{W}^{(\text{query})} \phi(\mathbf{t}) \right) \quad (5)$$

$$\mathbf{K} = \text{LayerNorm} \left(\mathbf{W}^{(\text{key})} \phi(\mathbf{X}) \right) \quad (6)$$

$$\mathbf{V} = \text{LayerNorm} \left(\mathbf{W}^{(\text{value})} \phi(\mathbf{X}) \right) \quad (7)$$

where $\mathbf{W}^{(\text{out})}$ is of shape $[d \times 2d]$, $\mathbf{W}^{(\text{proj})}$ is of shape $[d \times V]$ (with V the number of word types in our vocabulary), and all other matrices of shape $[d \times d]$; ϕ is a nonlinear activation function. The input \mathbf{X} corresponds to layer-normalized input embeddings for the words in the context of the attested word t , i.e., all tokens $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$ in the sentence except for t_i :

$$\mathbf{X} = \text{LayerNorm} \left(\begin{bmatrix} \mathbf{x}_{t_1} \\ \vdots \\ \mathbf{x}_{t_{i-1}} \\ \mathbf{x}_{t_{i+1}} \\ \vdots \\ \mathbf{x}_{t_n} \end{bmatrix} \right) \quad (8)$$

The position-aware model is highly similar to the position-agnostic model, except that we replace

eq. (5) with

$$\mathbf{q} = \text{LayerNorm} \left(\mathbf{W}^{(\text{query})} \phi(\mathbf{p}_i) \right) \quad (9)$$

and the input \mathbf{X} in eq. (8) is now defined as

$$\mathbf{X} = \text{LayerNorm} \left(\begin{bmatrix} \mathbf{x}_{t_1} + \mathbf{p}_1 \\ \vdots \\ \mathbf{x}_{t_{i-1}} + \mathbf{p}_{i-1} \\ \mathbf{x}_{t_{i+1}} + \mathbf{p}_{i+1} \\ \vdots \\ \mathbf{x}_{t_n} + \mathbf{p}_n \end{bmatrix} \right) \quad (10)$$

In detail, these models are centered on the use of a scaled-dot attention mechanism (Bahdanau et al., 2016; Vaswani et al., 2017) as shown in eq. (4): the hidden representation \mathbf{h} in eq. (4) is an average of the value representations in eq. (7), weighted by how similar key and query representations are (eqs. (5), (6) and (9)). Keys and values are computed from the context (eqs. (6) to (8) and (10)), whereas the query is derived from minimal input information about the term: In our position-aware architecture, this input is simply the index of the term (eq. (9)); in the position agnostic model, we use a default input vector \mathbf{t} for all terms, learned along with the other model parameters (eq. (5)).⁵ To further bolster the expressiveness of these language models, we include specific subnetworks linked to the computations of keys, values and queries, as well as a final computation block after the attention head (eq. (3)) and before projection onto the vocabulary space (eq. (2)).

As a useful reference point, we also include a word2vec CBOW model (Mikolov et al., 2013)—which, while not directly comparable, has been extensively studied in prior literature. For each model (including word2vec), we replicate training with three different seeds. Models are trained on a corpus of 20M sentences, half of which are sampled from Wikipedia, whereas the other half comes from BookCorpus (Zhu et al., 2015). Further details are available in Appendix A.

3.2 Accuracy

The first item we focus on is whether our models are accurate: How often is the most likely term according to $p(t|c)$ in fact the one we attest in our held out evaluation set?

⁵Using an attention mechanism allows us to dynamically weight the different value vectors based on the query and keys' vectors. This is therefore more expressive than the basic CBOW scheme of Mikolov et al. (2013), where all context items are always averaged with equal weights.

arch.	dataset	acc.	$\mathbb{E}[p(t c)]$
pos	bookcorpus	0.450 ± 0.001	0.346 ± 0.003
	wikipedia	0.397 ± 0.001	0.290 ± 0.003
nopus	bookcorpus	0.289 ± 0.001	0.200 ± 0.002
	wikipedia	0.193 ± 0.000	0.103 ± 0.002
w2v	bookcorpus	0.033 ± 0.000	0.003 ± 0.000
	wikipedia	0.033 ± 0.001	0.005 ± 0.000

Table 1: Model accuracy and mass assigned to the attested term (average of 3 runs).

Corresponding results are displayed in Table 1, which lists performances both in terms of accuracy (the proportion of terms ranked as first by the language model) and average probability assigned to the attested term t , noted $\mathbb{E}[p(t|c)]$. First, metrics on BookCorpus are always higher than their counterpart on Wikipedia—this likely stems from the higher average sentence length in the latter, along with the more diverse vocabulary it uses. None of the model pass the threshold of 50% accuracy, suggesting that most of the time, the most probable term (as ranked by our models) is not in fact the one we attest in the corpus. Second, we find a clear distinction between the three models considered: Word2vec fares significantly worse than the other two more complex models, but the addition of position also clearly improves both accuracy probability mass metrics as compared to the position-agnostic model. Third, we can see a fairly low standard deviation across all three runs—i.e., results are generally stable.

Overall, these results suggest a nuanced take: We do not find these models to be highly accurate, but we do see some confirmation of our hypothesis that linguistically informed context (in our case, positionally informed contexts) fare better.

3.3 Syntagmatic compatibility

It is however worth remembering that model accuracy is a flawed metric, and should not serve as a means of evaluating language models as models of the paradigmatic axis—since speakers and writers can and do elect to use unlikely terms. Instead, we ought to look at whether the words highlighted as relevant for a paradigm are compatible with the syntagmatic constraints of its context. As a simplified first step towards answering this, we consider looking at part of speech information: If the term we attest in our context is a noun, we should expect that the most likely terms according to $p(t|c)$

should all be nouns.⁶

A first technical question to solve, then, concerns how to establish *which set of likely terms* one should focus on: Given that paradigms retrieved from language models are probabilistic in nature, we need some means of deciding which words to rule in or out of a paradigmatic set. In practice, we need some manner of restricting the output vocabulary to the most likely terms. In the present work, we consider two simple approaches. The first consists in simply taking the top $k = 10$ most likely terms according to the model. The second, consists in using conformal prediction sets (CPS; Vladimir Vovk, 2005), a principled way of selecting a subset of the possible output terms so as to guarantee a coverage of $N = 80\%$. Simply put, a coverage of 80% entails that that selected subsets each have 80% chances of containing the attested term. In practice, we use a least-ambiguous set-valued classifier method (Sadinle et al., 2019): We (i) measure the probability mass assigned to each attested term on a held out calibration set; (ii) compute the $1 - N^{\text{th}}$ quantile q of these probability scores; and (iii) build sets from term distributions $p(t|c)$ by considering all values above that threshold quantile q , or $\mathcal{T} = \{t' : p(t'|c) \geq q\}$. Assuming symmetry and iid. between test and calibration data, the probability of the attested term t should be greater than q for $N\%$ of the test examples, and thus included in \mathcal{T} with a likelihood of $N\%$.

Having decided on how to select paradigm subsets, we can now turn to a second technical question: how to measure whether terms in a paradigm have the correct part-of-speech. POS-tagging systems that rely on full sentences to label words are not suitable to our purposes, since they could bias the labeling of terms in a paradigm towards the part-of-speech of the attested term by sheer virtue of the syntagmatic constraints of the context. Instead, our inquiry requires a context-independent means of establishing possible parts-of-speech for selected terms. We therefore fall back to a lexical resource—namely Wiktionary, owing to its large coverage;

⁶It is perhaps more common to evaluate distributional models on semantic tasks, given the distributional hypothesis expects contextual similarity to be linked to semantic similarity (Rubenstein and Goodenough, 1965; Miller and Charles, 1991; Hill et al., 2015). We depart from this tradition as this aspect of distributional representations seems to be somewhat consensual. While assessing the POS-tagging capabilities of language and embedding models alike has been studied extensively prior to this work (e.g., Elman, 1990; Lenci et al., 2022), little has been done to study whether the full output distribution of a language model is syntagmatically coherent.

method	arch.	dataset	% valid POS
baseline		bookcorpus	47.135
		wikipedia	47.385
CPS	pos	bookcorpus	87.250 ± 0.207
		wikipedia	84.087 ± 0.239
	nopos	bookcorpus	76.895 ± 0.198
		wikipedia	71.981 ± 0.093
	w2v	bookcorpus	60.337 ± 0.097
		wikipedia	60.775 ± 0.079
Top 10	pos	bookcorpus	81.929 ± 0.254
		wikipedia	80.490 ± 0.367
	nopos	bookcorpus	72.074 ± 0.180
		wikipedia	68.820 ± 0.233
	w2v	bookcorpus	71.961 ± 0.111
		wikipedia	71.551 ± 0.058

Table 2: Proportion of syntagmatically compatible likely paradigm terms, according to the POS tag of the attested term (average of 3 runs).

we rely on the English RDF parse by [Sérasset and Tchechmedjiev \(2014\)](#). This wide coverage, however, comes at the expense of leniency and accuracy. We therefore consider as a baseline using the full vocabulary as a paradigm subset: This gives us a strict lower bound for model performances. For simplicity, we ignore terms (both attested and in the paradigms) for which we find no Wiktionary entry; any term in a given paradigm is counted as syntagmatically compatible as long as one of its reported parts of speech could match one of the reported parts of speech of the attested term. We then report the average proportion of paradigm members that are syntagmatically compatible.

An overview of the corresponding results is displayed in Table 2. A few key observations need to be made. First, we can take notice of the very high lower bound suggested by our baseline—this can be explained in part by the leniency of our procedure as well as the noisiness of the POS-tag inventory derived from Wiktionary, although the categorical flexibility exhibited by the English lexicon may also play a role. We also highlight that all our experiments are clearly on average more compatible than this baseline—suggesting that, although our methodology suffers from its limitations, we can observe some evidence that the language modeling objective corresponds to establishing linguistically meaningful paradigms.

Furthermore, we see that terms in paradigms are generally more syntagmatically compatible

for BookCorpus paradigms rather than Wikipedia paradigms. This nuances our earlier discussions with respect to accuracy: Our language models appear indeed fundamentally less adequate when it comes to modeling paradigms in Wikipedia. A wider lexicon might entail a lesser ability to construct lexically meaningful representations of paradigmatic distributions: Exposing a language model to more numerous but rarer words might lower its average performance.

Lastly, we see that positional information significantly improves the syntagmatic compatibility of terms in paradigms. In a few cases, the word2vec baseline models are comparable to the position-agnostic language models. This hinges on the criterion used to establish paradigms: Selecting the top-10 highest probability scores yields less compatible sets than the quantile-based conformal set approach, except for word2vec. This should come as no surprise, given that the conformal sets are constructed based on the likelihood of an attested term. Word2vec models, as shown in Table 1, are generally not accurate in this regard; in particular, the probability mass they assign to the attested term tends to be low. Less accurate models therefore yield larger conformal sets, which we expect to be less syntagmatically compatible. This can be verified by looking at the average size of the conformal prediction sets: While the position-aware models yield conformal sets containing ≈ 42 terms in average, and the position agnostic ≈ 285 , this number rises to $\approx 26\,441$ for wordvec—i.e., more than a quarter of the vocabulary is included in the conformal set.

Sizes of the conformal prediction sets can interest us for another reason. We can expect that conformal prediction sets should be larger when paradigms can contain more words. In terms of parts-of-speech, we therefore expect that open grammatical categories like noun, verbs and adjectives should yield larger sets than closed categories, such as articles, conjunctions and prepositions.⁷ An overview of the CPS sizes, broken down per part-of-speech, is provided in Table 3, along with the number of relevant conformal sets. Open categories (verbs, nouns, proper nouns, adjectives) tend to yield the largest sets, whereas closed categories

⁷[Angelopoulos and Bates \(2022\)](#) suggest that conformal prediction set sizes can be used as proxies for model uncertainty: A larger conformal set is more ambiguous as to what the target should be. In short, we expect CPSs to capture the uncertainty inherent to the ambiguity of different parts of speech.

	number of CPSs	avg. CPS size		
		w2v	nopos	pos
adjective	106 908	26 386.5	306.1	46.1
adverb	91 442	26 897.5	297.0	34.7
article	27 229	26 693.2	294.5	21.3
conjunction	28 683	26 840.7	286.8	34.7
determiner	31 388	27 024.4	284.5	30.9
infix	7	26 917.3	314.4	71.0
interjection	30 691	26 765.7	287.9	35.6
noun	241 535	26 443.9	308.6	46.6
numeral	19 047	26 226.1	237.1	22.3
particle	27 708	26 855.1	208.4	19.0
phr. unit	6563	26 832.1	294.3	28.7
postposition	868	26 596.1	316.8	47.9
prefix	1400	26 721.6	82.1	8.2
preposition	80 831	26 824.7	291.5	27.3
pronoun	45 210	26 922.3	277.7	28.7
proper noun	13 776	26 489.7	303.4	42.6
suffix	19 321	26 583.3	297.6	26.7
symbol	19 906	26 414.7	282.0	24.4
verb	159 314	26 489.1	319.0	51.0
all	354 388	26 440.6	285.0	42.0

Table 3: Conformal prediction sets size per part of speech (averages of 3 runs).

(aside from the two least represented, infixes and postpositions) yield smaller conformal prediction sets. In fact, the difference in CPSs sizes between nouns, verbs, adjectives, adverbs and proper nouns vs. those for all other parts of speech is statistically significant.⁸

4 Conclusion

In the present article, we have argued that language models and word embeddings can be understood through a structuralist lens as models of the paradigmatic axis, as long as we factor in the inherent ambiguous nature of language modeling objectives. We have highlighted how this conception builds upon prior work (Sahlgren, 2008; Gastaldi, 2021; Gastaldi and Pellissier, 2021), and where it distinguishes itself from these prior approaches—in terms of the range of models it considers, as well as by explicitly embracing the departures from the earlier formulations of this structuralist concept. The position we endorse here is to minimize the assumptions necessary to frame language models in structuralist terms: With fewer assumptions comes broader application. In contrast, Gastaldi

⁸Mann-Whitney U tests: $p < 10^{-32}$, common-language effect size $f > 0.66$ in position-aware models and $f > 0.57$ in position-agnostic models

and Pellissier’s (2021) position can be understood as a narrower form of the present argument designed to allow the emergence of structural representations of the context—but it is worth asking whether one should really expect of distributional models that they yield explicit structural representations (Rumelhart and McClelland, 1986; Buder-Gröndahl, 2023).



One crucial point we have left out of our discussion concerns whether purely linguistic paradigms actually exist. The data we use to train distributional models are not in fact linguistic in nature, but sociolinguistic; they encode social variation and biases, and consequently distributional models do as well (Bolukbasi et al., 2016; Garg et al., 2018). We should expect the paradigms that the language modeling objective obtains to not purely encode linguistic relationships. As such, it is crucial to evaluate the extent to which we can abstract away from the sociolinguistic aspect of the training data.

Hence, one contribution of the present work is to propose a preliminary empirical verification of whether this conception of language models (and therefore word embeddings) as models of the paradigmatic axis is coherent. To that end, we have demonstrated how manipulating the linguistic information in the input contexts of conceptually simple architectures yields predictable effects, and how conformal prediction sets can be leveraged to select paradigm terms in a linguistically meaningful way—in that selected terms are syntagmatically compatible with the context from which we derive them.

In the present work, we have striven to provide a basis that is easy to comprehend and straightforward to build upon—which comes at the cost of our experiments and models being simplistic in many regards. This work also leaves a number of research questions open for future inquiries: Do larger models yield more accurate representations of the paradigmatic axis? What other linguistic information should we include or remove from our contexts? How do these models behave with respect to other pre-segmentations of the training corpora—and especially the ubiquitous word-piece segmentations? How can a model of the paradigmatic axis be leveraged to study other linguistic phenomena, and what methodological steps should we take to mitigate its potential lack of accuracy?

Acknowledgements

We thank Timothée Bernard, Tommi Buder-Gröndahl, Mathilde Huguin, Jussi Karlgren and Denis Paperno, as well as the three anonymous reviewers for discussions and comments on this work that substantially bettered it.

 This work is part of the FoTran project, funded by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation program (agreement N° 771113).  We also thank the CSC-IT Center for Science Ltd., for computational resources. This work is also supported by the ICT 2023 project “Uncertainty-aware neural language models” funded by the Academy of Finland (grant agreement N° 345999).

References

- Anastasios N. Angelopoulos and Stephen Bates. 2022. A gentle introduction to conformal prediction and distribution-free uncertainty quantification.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.
- Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. 2014. Frege in space: A program for composition distributional semantics. *Linguistic Issues in Language Technology*, 9.
- 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.
- Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.
- Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Adam Bouyamourn. 2023. Why LLMs hallucinate, and how to get (evidential) closure: Perceptual, intensional, and extensional learning for faithful natural language generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3181–3193, Singapore. Association for Computational Linguistics.
- Mikael Brunila and Jack LaViolette. 2022. What company do words keep? Revisiting the distributional semantics of J.R. Firth & Zellig Harris. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4403–4417, Seattle, United States. Association for Computational Linguistics.
- Tommi Buder-Gröndahl. 2023. The ambiguity of BERTology: what do large language models represent? *Synthese*, 203(1):15.
- Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.
- Noam Chomsky. 1965. *Aspects of the Theory of Syntax*, 50 edition. The MIT Press.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.
- Guy Emerson. 2020. What are the goals of distributional semantics? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7436–7453, Online. Association for Computational Linguistics.
- Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.
- Ananya Ganesh, Jie Cao, E. Margaret Perloff, Rosy Southwell, Martha Palmer, and Katharina Kann. 2023. Mind the gap between the application track and the real world. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1833–1842, Toronto, Canada. Association for Computational Linguistics.

- Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. [Word embeddings quantify 100 years of gender and ethnic stereotypes](#). *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.
- Juan Luis Gastaldi. 2021. [Why can computers understand natural language?](#) *Philosophy & Technology*, 34(1):149–214.
- Juan Luis Gastaldi and Luc Pellissier. 2021. The calculus of language: explicit representation of emergent linguistic structure through type-theoretical paradigms. *Interdisciplinary Science Reviews*, 46(4):569–590.
- Zellig Harris. 1954. [Distributional structure](#). *Word*, 10(2-3):146–162.
- Dan Hendrycks and Kevin Gimpel. 2016. [Gaussian error linear units \(GELUs\)](#).
- Aurélie Herbelot and Ann Copestake. 2021. [Ideal words](#). *KI - Künstliche Intelligenz*, 35(3):271–290.
- John Hewitt and Christopher D. Manning. 2019. [A structural probe for finding syntax in word representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. [SimLex-999: Evaluating semantic models with \(genuine\) similarity estimation](#). *Computational Linguistics*, 41(4):665–695.
- Louis Hjelmslev. 1971. *Prolégomènes à une théorie du langage. suivi de "La structure fondamentale du langage"*. Éditions de Minuit.
- Thomas K. Landauer and Susan T. Dumais. 1997. [A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge](#). *Psychological Review*, 104(2):211–240.
- Alessandro Lenci. 2018. [Distributional models of word meaning](#). *Annual Review of Linguistics*, 4(1):151–171.
- Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2022. [A comparative evaluation and analysis of three generations of distributional semantic models](#). *Lang. Resour. Eval.*, 56(4):1269–1313.
- Omer Levy and Yoav Goldberg. 2014. [Neural word embedding as implicit matrix factorization](#). In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- 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.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Jacqueline Léon. 2011. [Z. S. Harris and the semantic turn of mathematical information theory](#). In *History of Linguistics 2008*, pages 449–458. John Benjamins.
- Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2017. [Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation](#). *Journal of Memory and Language*, 92:57–78.
- Louise McNally. 2017. *Kinds, descriptions of kinds, concepts, and distributions*, pages 39–62. Düsseldorf university press, Berlin, Boston.
- William Merrill, Alex Warstadt, and Tal Linzen. 2022. [Entailment semantics can be extracted from an ideal language model](#). In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 176–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). *arXiv preprint arXiv:1301.3781*.
- George A. Miller and Walter G. Charles. 1991. [Contextual correlates of semantic similarity](#). *Language and Cognitive Processes*, 6(1):1–28.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21(1).
- Herbert Rubenstein and John B. Goodenough. 1965. [Contextual correlates of synonymy](#). *Commun. ACM*, 8(10):627–633.
- David E Rumelhart and James L McClelland. 1986. [On learning the past tenses of english verbs](#).
- Mauricio Sadinle, Jing Lei, and Larry Wasserman. 2019. [Least ambiguous set-valued classifiers with bounded error levels](#). *Journal of the American Statistical Association*, 114(525):223–234.
- Magnus Sahlgren. 2008. [The distributional hypothesis](#). *The Italian Journal of Linguistics*, 20:33–54.
- Gerard Salton, Anita Wong, and Chun-Shu Yang. 1975. [A vector space model for automatic indexing](#). *Commun. ACM*, 18(11):613–620.
- Ferdinand de Saussure. 1916. *Cours de linguistique générale*. Payot, Paris.
- Hinrich Schütze. 1992. [Word space](#). In *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann.

Gilles Sérasset and Andon Tehechmedjiev. 2014. [Dbnary : Wiktionary as Linked Data for 12 Language Editions with Enhanced Translation Relations](#). In *3rd Workshop on Linked Data in Linguistics: Multilingual Knowledge Resources and Natural Language Processing*, pages 67–71, Reykjavik, Iceland.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. [Masked language modeling and the distributional hypothesis: Order word matters pre-training for little](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rich Sutton. 2019. [The bitter lesson](#).

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajič, Daniel Hershcovich, Eduard Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, Ekaterina Shutova, and Roberto Navigli. 2023. [What’s the meaning of superhuman performance in today’s NLU?](#) In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, Toronto, Canada. Association for Computational Linguistics.

Jaap van Marle. 1984. *On the Paradigmatic Dimension of Morphological Creativity*. Foris Publications, Dordrecht, The Netherlands.

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*, volume 30. Curran Associates, Inc.

Glenn Shafer Vladimir Vovk, Alexander Gammerman. 2005. *Algorithmic Learning in a Random World*. Springer-Verlag.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. [Aligning books and movies: Towards story-like visual explanations by watching movies and reading books](#). In *The IEEE International Conference on Computer Vision (ICCV)*.

A Implementation details

For the position-aware and position-agnostic models, we use a latent dimension of $d = 256$ and a GELU activation function (Hendrycks and Gimpel, 2016). We optimize cross-entropy between the model output and the attested term at each position, using the Adam optimization algorithm with decoupled weight decay (Loshchilov and Hutter, 2019), using a learning rate of 0.001, $\beta = (0.9, 0.999)$, and a weight decay of 0.01.

Models are trained on a corpus of 20M sentences, half of which are sampled from Wikipedia, whereas

the other half comes from BookCorpus (Zhu et al., 2015): These corpora corresponds to the sources used for training BERT (Devlin et al., 2019), but the amount of data we consider here is orders of magnitude lower. We also select 20k sentences for testing, and 2k for further calibration in Section 3.3; likewise, half of the sentences in both sets are sampled from Wikipedia and half from BookCorpus. We pre-segment the corpus in words using nltk (Bird and Loper, 2004), using a vocabulary comprising the 100k most frequent words; we pre-process all sentences by lowercasing, stripping accents, and normalizing to the NFKD unicode norm. Models are trained for one epoch over these data, by minibatches of 50 sentences truncated to a maximum length of 128 tokens.

The word2vec baselines are trained on the same data using a vector size of 100, window of 5, and 5 negative examples per target. For our language models, training requires 12 to 16h hours on a RTX 3080 GPU, and about half an hour on CPUs for the word2vec baseline.

A Generalized Algorithm for Learning Positive and Negative Grammars with Unconventional String Models

Sarah Payne

Stony Brook University
sarah.payne@stonybrook.edu

Abstract

This paper introduces an algorithm for learning positive and negative grammars with enriched representational models. In conventional model-theoretic treatments of strings, each position belongs to exactly one unary relation. Strother-Garcia et al. (2016) introduce unconventional string models, in which multiple positions can have shared properties, and demonstrate their utility for grammatical inference. Chandlee et al. (2019) develop this approach for learning negative grammars. Here, we show that by fixing k — the size of the elements in the grammar — Chandlee et al.’s approach can be further generalized to learn both positive and negative grammars over unconventional string models. We prove that this algorithm finds the most general grammars which cover the data.

1 Introduction

A great deal of work on learning formal languages has made use of **conventional string models**, in which each position in a string belongs to exactly one unary relation (Heinz, 2010b; Heinz et al., 2012, i.a.). In this paper, we focus on learning over **unconventional string models**, in which positions in a string can have multiple, shared properties (§2.3; Strother-Garcia et al., 2016; Vu et al., 2018). For phonological applications, we can think of conventional string models as operating exclusively over segments — atomic, undecomposable, units — while unconventional string models operate over phonological features.

We focus on the learning of formal languages that can be defined by a set of banned (under a **negative** grammar, Rogers et al. 2013) or allowed (under a **positive** grammar, Heinz et al. 2012) substructures. These include the Strictly k -Local and Strictly k -Piecewise classes (Rogers et al., 2010; Rogers and Pullum, 2011, i.a.); many phonological and phonotactic generalizations fall into these classes (e.g., Heinz, 2018). While Strother-Garcia

et al. (2016), Chandlee et al. (2019), and Rawski (2021) develop algorithms for learning negative grammars with unconventional string models, how to learn positive grammars with these models remains an open question. Recent work on language acquisition suggests that the child may construct positive phonological (Belth, 2023) and phonotactic (Payne, 2023) grammars, in line with evidence for positive syntactic and morphological grammars (e.g., Marcus et al., 1992; Yang, 2016; Belth et al., 2021; Li and Schuler, 2023). While arguments have also been made for negative phonological grammars (e.g., Prince and Smolensky, 1993; Hayes and Wilson, 2008), these findings demonstrate that the learning of positive grammars from unconventional string models warrants further exploration.

When learning over conventional string models, positive and negative grammars are straightforwardly interdefinable (Heinz, 2010b); we may learn a positive grammar simply by learning a negative one and applying a post-hoc conversion. However, such a conversion is exponentially more expensive for unconventional string models (§4.3). What’s more, the polarity of the grammar to be learned has implications for the learning trajectory: while the language of the grammar continuously *shrinks* as a negative grammar grows, it continuously *expands* as a positive grammar grows (§8). Hence, there exist independent psycholinguistic and computational motivations for learning positive grammars directly from unconventional string models.

In this paper, we adapt the learning algorithm of Chandlee et al. (2019) to learn both positive and negative grammars over unconventional string models. Specifically, Chandlee et al. exploit the partially-ordered hypothesis space given by unconventional string models to learn the most general *negative* grammars. We demonstrate that if the size of substructures in the grammar is fixed to be exactly k , then we can immediately adapt this algorithm to learn both the most general positive and

negative grammars. What’s more, for any negative grammar learnable by the [Chandlee et al.](#) algorithm, our algorithm learns an equivalent negative grammar (§4.2). This paper is organized as follows: §2 provides preliminaries of model theory, §3 introduces subfactors and maxfactors, and §4 defines positive and negative grammars and their languages in terms of these structures. §5 defines the learning criteria, adapted from [Chandlee et al.](#), and §6 introduces a generalized learning algorithm that provably satisfies these criteria. The algorithm is applied to the example of Samala sibilant harmony in §7 and implications are discussed in §8.

2 Preliminaries

This section and the next follow closely from §2-3 of [Chandlee et al. \(2019\)](#), since the current work builds closely on their algorithm.

2.1 Formal Language Theory

Formal language theory allows us to study languages as mathematical objects which exist independently of the specific grammar ([Heinz, 2016](#)). The set of all possible finite strings generated from a finite alphabet Σ is denoted Σ^* , and the set of all strings of length k is given by Σ^k . In formal language theory, languages are defined as subsets of Σ^* . The length of a string w is denoted $|w|$.

2.2 Finite Model Theory

Finite model theory provides a unified vocabulary for representing many kinds of objects as relational structures, allowing for algorithms that are largely agnostic to the choice of linguistic representation ([Enderton, 2001](#); [Libkin, 2004](#); [Chandlee et al., 2019](#); [Lambert et al., 2021](#); [Rawski, 2021](#)). We consider finite relational models of strings in Σ^* .

Definition 1 (Models). A **model signature** is a set of relations $R = \{R_1, R_2, \dots, R_n\}$ where each R_i is an m_i -ary relation. An **R-structure** is a tuple of elements $S = \langle D; R_1, R_2, \dots, R_n \rangle$, where D is a finite set of elements, the domain, and each R_i is a subset of D^{m_i} . The **size** $|S|$ of an R-structure S corresponds to the cardinality of its domain. A **model** for the set of objects Ω is a total, one-to-one function from Ω to R-structures.

Consider the **precedence** model for strings in Σ^* , defined as $M^<(w) := \langle D^w; <, [R_\sigma^w]_{\sigma \in \Sigma} \rangle$ where $D^w = \{1, \dots, |w|\}$ is the domain of positions in w and $< := \{(i, j) \in D^w \times D^w \mid i < j\}$ is the general precedence relation ([Büchi, 1960](#); [McNaughton](#)

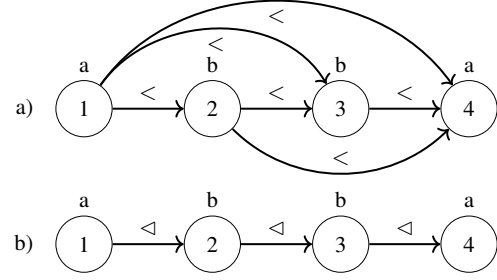


Figure 1: The precedence ($M^<$, subfigure a) and successor ($M^<$, subfigure b) models of the string $abba$.



Figure 2: A visualization of the R-structure given by $S_{ab,ba} = \langle D = \{1, 2, 3, 4\}; < = \{(1, 2), (3, 4)\}, R_a = \{1, 4\}, R_b = \{2, 3\}, R_c = \emptyset \rangle$.

and [Papert, 1971](#); [Rogers et al., 2013](#)). With this model and $\Sigma = \{a, b, c\}$, we have:

$$\begin{aligned} M^<(abba) &= \langle D = \{1, 2, 3, 4\}; \\ &< = \{(1, 2), (1, 3), (1, 4), \\ &(2, 3), (2, 4), (3, 4)\}, \\ &R_a = \{1, 4\}, R_b = \{2, 3\}, R_c = \emptyset \rangle \end{aligned} \quad (1)$$

The **successor** model differs from the precedence model only in the ordering relation, given by $< := \{(i, i + 1) \in D^w \times D^w\}$. The precedence and successor models of $abba$ are shown in Figure 1.

Since R-structures may be *any* mathematical structure conforming to a model signature, not all possible R-structures are valid models of strings: the R-structure in Figure 2 is not a model of any $w \in \Sigma^*$. To limit the R-structures we consider, we introduce the notion of **connectedness**.

Definition 2 (Connected R-Structure). An R-structure $S = \langle D; R_1, R_2, \dots, R_n \rangle$ is connected iff $(\forall x, y \in D)[(x, y) \in C^*]$, where C^* is defined as the symmetric transitive closure of:

$$\begin{aligned} C &= \{(x, y) \in D \times D \mid \\ &\exists i \in \{1 \dots n\}, \exists (x_1 \dots x_m) \in R_i \\ &\exists s, t \in \{1 \dots m\}, x = x_s, y = x_t\} \end{aligned} \quad (2)$$

Intuitively, domain elements x and y of S belong to C if they belong to some non-unary relation R_i in S . It is easy to see that $S_{ab,ba}$ (Figure 2) is not connected: neither $(2, 3)$ nor $(3, 2)$ is contained in

C and thus none of (1,3), (1,4), (2,3), (2,4), etc. are contained in C^* . In contrast, both $M^{\langle}(abba)$ and $M^{\triangleleft}(abba)$ in Figure 1 are connected R-structures.

2.3 Unconventional String Models

The models shown in Figure 1 are **conventional string models**: besides the ordering relation, they include only mutually-exclusive unary relations (e.g., R_a) which label each domain element with a single property of being some $\sigma \in \Sigma$. In contrast, **unconventional string models** recognize that distinct alphabetic symbols may share properties and expand the model signature by including these properties as non-exclusive unary relations (Strother-Garcia et al., 2016; Vu et al., 2018).

Unconventional string models allow for more generalized representations, and thus have a number of useful linguistic applications. Consider the example of sibilant harmony in Samala: subsequences such as [s...s] that agree in \pm ANTERIOR are allowed but subsequences such as [s...ʃ] which disagree are banned, so [hasxintilawas] is licit but [hasxintilawaf] is not (Hansson, 2010). Under a conventional string model, we must separately represent that [s...ʃ], [z...ʃ], [s...ʒ], etc. are banned, or equivalently that [s...s], [z...z], [ʃ...ʒ], etc. are allowed. Under an unconventional string model, however, we can simply represent that [+STR, +ANT][+STR, -ANT] subsequences are banned, or that [+STR, +ANT][+STR, +ANT] and [+STR, -ANT][+STR, -ANT] subsequences are allowed.

3 Subfactors and Maxfactors

We define a partial order over R-structures by establishing the notions of **restrictions**, **subfactors**, and **maxfactors**, building on Chandlee et al. (2019).

Definition 3 (Restriction). An R-structure A is a restriction of an R-structure B if $D^A \subseteq D^B$ and for each m -ary relation R_i in the model signature, $R_i^A = \{(x_1, \dots, x_m) \in R_i^B \mid x_1, \dots, x_m \in D^A\}$.

A restriction is made by identifying a subset D^A of the domain of B and retaining only those relations in B whose elements are wholly within D^A . For example, Figure 3 shows the restriction of $M^{\langle}(abba)$ as defined in Equation 1 to $D' = \{1, 2, 3\}$. This restriction is given by: $M^{\langle}(abb) = \langle D' = \{1, 2, 3\}; < = \{(1, 2), (1, 3), (2, 3)\}, R_a = \{1\}, R_b = \{2, 3\}, R_c = \emptyset \rangle$.

Definition 4 (Subfactor). A connected R-structure A is a **subfactor** of an R-structure B (notated

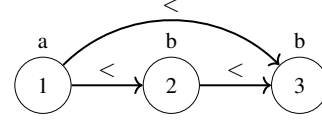


Figure 3: A restriction of $M^{\langle}(abba)$ shown in Fig. 1a.

$A \sqsubseteq B$) if there exists a restriction B' of B and a bijection h such that for all $R_i \in R$, if $R_i(x_1, \dots, x_m)$ holds in A , then $R_i(h(x_1), \dots, h(x_m))$ holds in B' . If $A \sqsubseteq B$, B is a **superfactor** of A .

Definition 5 (Maxfactor). A connected R-structure A is a **maxfactor** of an R-structure B (notated $A \leq B$) iff $A \sqsubseteq B$ and for each m -ary relation R_i , whenever $R_i(x_1, \dots, x_m)$ holds in B , $R_i(h^{-1}(x_1), \dots, h^{-1}(x_m))$ holds in A . Equivalently, $A \leq B$ if $A \sqsubseteq B$ and there is no R-structure A' non-isomorphic to A and B such that $|A| = |A'|$ and $A \sqsubseteq A' \sqsubseteq B$.¹

Intuitively, A is a subfactor of B if there is a mapping between D^A and some subset of D^B and all relations that hold in A also hold over the corresponding elements in B . Note that this requirement is *unidirectional*. By contrast, maxfactors additionally require that all relations that hold in B also hold over the corresponding elements in A . We can thus think of maxfactors as the *maximally specified* subfactors of an R-structure. We use **factor** when the distinction between subfactor and maxfactor is irrelevant. This is true for conventional string models: since there is no underspecification in these models, any subfactor must also be a maxfactor.

If $A \sqsubseteq B$ and $|A| = k$, then A is a **k -subfactor** of B , and if $A \leq B$ and $|A| = k$, then A is **k -maxfactor** of B . Let the set of k -subfactors of an R-structure B be given by:

$$\text{SFAC}_k(B) := \{A \mid A \sqsubseteq B, |A| = k\} \quad (3)$$

and the set of k -maxfactors of B be given by:

$$\text{MFAC}_k(B) := \{A \mid A \leq B, |A| = k\} \quad (4)$$

For all $w \in \Sigma^*$ and any model M of Σ^* , the k -subfactors and k -maxfactors of w are given by $\text{SFAC}_k(M(w))$ and $\text{MFAC}_k(M(w))$, respectively; we also write $\text{SFAC}_k(M, w)$ and $\text{MFAC}_k(M, w)$ for readability. Finally, we define:

$$\text{SFAC}_k(M, \Sigma^*) = \bigcup_{w \in \Sigma^*} \text{SFAC}_k(M, w) \quad (5)$$

¹In model-theoretic terms, Definition 5 simply means that A is a connected substructure of B (Libkin, 2004).

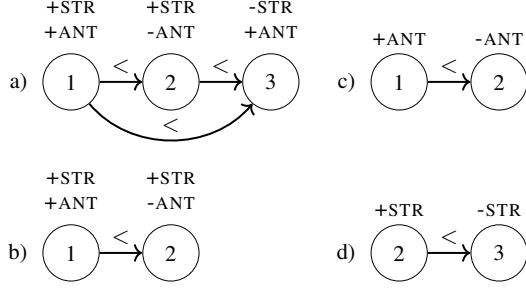


Figure 4: A visualization of $M^<(\text{fst})$ (a) and three 2-subfactors; only subfactor (b) is also a 2-maxfactor.

and likewise for $\text{MFAC}_k(M, \Sigma^*)$. From Definitions (4) and (5), we have:

$$\text{MFAC}_k(M, w) \subseteq \text{SFAC}_k(M, w) \quad (6)$$

Note that our definition of $\text{SFAC}_k()$ differs from the that of Chandlee et al. (2019) in that Chandlee et al. require the size of the subfactors to be bounded by k rather than equal to k . This difference is due to the constant size needed to define a positive grammar, discussed in §4.1. To differentiate between our definition of $\text{SFAC}_k()$ and that of Chandlee et al. (2019), we denote the latter as:

$$\text{SFAC}_{\leq k}(B) := \{A \mid A \subseteq B, |A| \leq k\} \quad (7)$$

Returning to Samala sibilant harmony (§2.3), consider the precedence model of [sft] given by:

$$\begin{aligned} M^<(\text{sft}) &= \langle D = \{1, 2, 3\}; \\ &\leq = \{(1, 2), (1, 3), (2, 3)\}, \\ R_{\text{STR}} &= \{1, 2\}, R_{\text{ANT}} = \{1, 3\} \end{aligned} \quad (8)$$

This model, along with three of its 2-subfactors, is shown in Figure 4. The R-structures (b), (c), and (d) are subfactors of $M^<(\text{sft})$, since they are all connected and the relations that hold within them also hold in $M^<(\text{sft})$. However, *only* (b) is a 2-maxfactor: it is the only subfactor for which all relations that hold in $M^<(\text{sft})$ also hold within it.

We now introduce two lemmas that will be used to define grammars and their languages in §4.

Lemma 1 (Maxfactor-Subfactor Containment). *Let k be some positive integer and let M be some model of Σ^* . For any $w \in \Sigma^*$ and for any $F \in \text{SFAC}_k(M, w)$, we have that:*

$$[\exists G \in \text{MFAC}_k(M, w)](F \sqsubseteq G) \quad (9)$$

Proof. Let G be the restriction of $M(w)$ to $h(D^F)$, where $h : F \rightarrow M(w)$ is given by Definition 4.

Clearly, $G \sqsubseteq M(w)$, and by Definition 3, $R_i^G = \{(x_1, \dots, x_m) \in R_i^{M(w)} \mid x_1, \dots, x_m \in h(D^F)\}$ for all $R_i \in R$. Thus, $G \sqsubseteq M(w)$ and whenever $R_i(x_1, \dots, x_m)$ holds in $M(w)$, $R_i(x_1, \dots, x_m)$ holds in G . By Definition 5, $G \leq M(w)$.

By Definition 4, for all $R_i \in R$, if $R_i(x_1, \dots, x_m)$ holds in F , then $R_i(h(x_1), \dots, h(x_m))$ holds in some restriction M' of $M(w)$, and thus in $M(w)$ by Definition 3. But since G is defined over $h(D^F)$ and contains all relations in $M(w)$ defined over D^G , it must also be the case that if $R_i(x_1, \dots, x_m)$ holds in F , then $R_i(h(x_1), \dots, h(x_m))$ holds in G . By Definition 4, this means that $F \sqsubseteq G$. \square

Lemma 2 (Union of Subfactors of Maxfactors). *Let k be some positive integer and let M be some model of Σ^* . For any $w \in \Sigma^*$, we have that:*

$$\bigcup_{S \in \text{MFAC}_k(M, w)} \text{SFAC}_k(S) = \text{SFAC}_k(M, w) \quad (10)$$

Proof. (\subseteq) Consider some $f \in \text{SFAC}_k(S)$, where $S \in \text{MFAC}_k(M, w) \subseteq \text{SFAC}_k(M, w)$ and thus $f \sqsubseteq S \in \text{SFAC}_k(M, w)$. By Equation 3, this means that $f \sqsubseteq S \sqsubseteq M(w)$, and thus that $f \sqsubseteq M(w)$. Since $f \sqsubseteq M(w)$ and $|f| = k$, by Equation 4, $f \in \text{SFAC}_k(M, w)$.

(\supseteq) Consider some $g \in \text{SFAC}_k(M, w)$. By Lemma 1, $[\exists g' \in \text{MFAC}_k(M, w)](g \sqsubseteq g')$, and since $g \sqsubseteq g'$ and $|g| = k$, Equation 3 tells us that $g \in \text{SFAC}_k(g')$. Since $g \in \text{SFAC}_k(g')$ and $g' \in \text{MFAC}_k(M, w)$, it must be the case that $g \in \bigcup_{S \in \text{MFAC}_k(M, w)} \text{SFAC}_k(S)$. \square

4 Grammars and Their Languages

4.1 Positive vs. Negative Interpretations

We define a grammar G as a finite set of subfactors; the language that G defines differs based on whether we interpret it as a positive or negative grammar. We first discuss these interpretations informally, then formalize them in Definitions 6-7.

Under a **negative** interpretation (notated G^-), the elements of G^- are forbidden, and strings in $L(G^-)$ contain no forbidden subfactors. This approach has parallels to logical expressions which are "conjunctions of negative literals" (Rogers et al., 2013; Chandlee et al., 2019): the forbidden subfactors are simply interpreted as the negative literals. Returning to Samala sibilant harmony (Figure 4), if (c) is in the grammar (i.e., $[+\text{ANT}][-\text{ANT}] \in G^-$), this is sufficient to determine that $\text{sft} \notin L(G^-)$, since $[+\text{ANT}][-\text{ANT}] \sqsubseteq M^<(\text{sft})$ (Figure 4a).

Under a **positive** interpretation (notated G^+), the elements of G^+ are permissible, and strings in $L(G^+)$ are those which are *covered* by these elements; we can think of the subfactors in G^+ as *tiling* the strings in $L(G^+)$ (Rogers and Heinz, 2014). Returning to Figure 4, if both (c) and (d) are in G^+ (i.e., $[+\text{STR}][-\text{STR}] \in G^+$, $[+\text{ANT}][-\text{ANT}] \in G^+$), then $\text{sft} \in L(G^+)$, since (c) covers $[\text{sf}]$ and (d) covers $[\text{ft}]$. However, if (d) but not (c) is in G^+ , then $\text{sft} \notin L(G^+)$, because there is *no subfactor* in G^+ that covers the first two indices of $M^<(\text{sft})$. The notion of tiling is greatly simplified when the subfactors used to tile are of equal size, as in Rogers and Heinz (2014). As such, our definitions of negative and positive grammars and their languages below operate over fixed values of k .

Definition 6 (Negative Grammar). Let k be some positive integer and M be a model of Σ^* . A **negative grammar** G^- is a subset of $\text{SFAC}_k(M, \Sigma^*)$, and the language $L(G^-)$ of G^- is given by:

$$L(G^-) = \{w \in \Sigma^* \mid (\forall S \in \text{MFAC}_k(M, w)) [\text{SFAC}_k(S) \cap G^- = \emptyset]\} \quad (11)$$

or equivalently by:

$$L(G^-) = \{w \in \Sigma^* \mid (\nexists S \in \text{MFAC}_k(M, w)) [\text{SFAC}_k(S) \cap G^- \neq \emptyset]\} \quad (12)$$

The class of such languages is defined as:

$$\mathcal{L}^-(M, k) = \{L \mid (\exists G^- \subseteq \text{SFAC}_k(M, \Sigma^*)) [L(G^-) = L]\} \quad (13)$$

Definition 7 (Positive Grammar). Let k be some positive integer and M be a model of Σ^* . A **positive grammar** G^+ is a subset of $\text{SFAC}_k(M, \Sigma^*)$, and the language $L(G^+)$ of G^+ is given by:

$$L(G^+) = \{w \in \Sigma^* \mid (\forall S \in \text{MFAC}_k(M, w)) [\text{SFAC}_k(S) \cap G^+ \neq \emptyset]\} \quad (14)$$

or equivalently by:

$$L(G^+) = \{w \in \Sigma^* \mid (\nexists S \in \text{MFAC}_k(M, w)) [\text{SFAC}_k(S) \cap G^+ = \emptyset]\} \quad (15)$$

The class of such languages is defined as:

$$\mathcal{L}^+(M, k) = \{L \mid (\exists G^+ \subseteq \text{SFAC}_k(M, \Sigma^*)) [L(G^+) = L]\} \quad (16)$$

Defining the languages of positive and negative grammars in terms of quantification over $\text{MFAC}_k(M, w)$ allows us to tile a word w with k -subfactors. We can think of Equations 11 through

	$\in G$	$\notin G$
\forall	Positive Grammar (Equation 14)	Negative Grammar (Equation 11)
\exists	Negative Grammar (Equation 12)	Positive Grammar (Equation 15)

Figure 5: Positive and negative grammars and their languages, organized by quantification and attestation.

15 as realizing two primary distinctions: **quantification** (\forall vs. \exists) and **membership** in G . For universal quantification (\forall), if all k -maxfactors of w are superfactors of some k -subfactor in G^+ , then $w \in L(G^+)$ (Equation 14), and if all k -maxfactors of w are *not* superfactors of some k -subfactor in G^- , then $w \in L(G^-)$ (Equation 11). For existential quantification (\exists), if there exists a k -maxfactor of w that is a superfactor of some k -subfactor in G^- , then $w \notin L(G^-)$ (Equation 12). If there exists a k -maxfactor of w that is *not* a superfactor of some k -subfactor in G^+ , then $w \notin L(G^+)$ (Equation 15). Figure 5 illustrates how these distinctions define the languages of positive and negative grammars.

To further illustrate the differences between negative and positive grammars, consider a grammar G such that $L(G) = \Sigma^*$, and let $k = 1$. If G is positive, then it must contain subfactors of *all possible* 1-maxfactors of any $w \in \Sigma^*$. The empty 1-subfactor $[\]$ satisfies this, so we have $G^+ = \{[\]\}$ and $L(G^+) = \Sigma^*$. Conversely, if we have $G^- = \{[\]\}$, we will have $L(G^-) = \emptyset$, since there is no $w \in \Sigma^*$ whose 1-maxfactors are *not* superfactors of $[\]$. To define a negative grammar accepting Σ^* , we must instead ensure that *no possible* 1-maxfactor of any $w \in \Sigma^*$ is a superfactor of an element in G^- ; this is easily achieved with $G^- = \emptyset$. At the same time, if we have $G^+ = \emptyset$, then no word $w \in \Sigma^*$ will have its 1-maxfactors contained by elements in G^+ , and thus $L(G^+) = \emptyset$.

4.2 Equivalence to Chandlee et al. (2019)

In contrast to the current work, Chandlee et al. (2019) focus only on the learning of negative grammars, defined as follows:

Definition 8 (Chandlee et al. Negative Grammar). Let k be some positive integer and M be a model of Σ^* . A negative grammar G^- is a subset of $\text{SFAC}_{\leq k}(M, \Sigma^*)$, and the language $L(G^-)$ is given by:

$$L(G^-) = \{w \in \Sigma^* \mid \text{SFAC}_{\leq k}(M, w) \cap G^- = \emptyset\} \quad (17)$$

Since [Chandlee et al.](#) consider negative grammars, they need only to set an *upper bound* on k . To learn both positive and negative grammars, however, we fix k . We thus wish to demonstrate an equivalence between the grammars and languages in Definition 8 and in Definition 6, namely:

Theorem 1. *Let L_{17} refer to $L(G^-)$ as in Equation (17) and L_{11} refer to $L(G^-)$ as in Equation (11). For any $G_1^- \subseteq \text{SFAC}_{\leq k}(M, \Sigma^*)$, $\exists G_2^- \subseteq \text{SFAC}_k(M, \Sigma^*)$ such that $L_{17}(G_1^-) = L_{11}(G_2^-)$.*

A full proof is provided in Appendix A.

4.3 The Cost of Interdefinability

When defined over conventional string models, negative and positive grammars are straightforwardly interdefinable: $G^+ = \Sigma^k \setminus G^-$ and $G^- = \Sigma^k \setminus G^+$ ([Heinz, 2010b](#)). However, there are two complications for unconventional string models that make the interdefinability significantly more costly.

Firstly, the number of potential k -subfactors is significantly larger for unconventional string models: consider a model with n binary features, defining $s \leq 2^n$ segments. Under a conventional string model, the number of k -factors is no more than $(s)^k \leq (2^n)^k$, since one segment is chosen at each position ([Heinz, 2010b](#)). Under an unconventional string model, however, a feature can be either positive, negative, or *underspecified* at each position, yielding $(3^n)^k$ possible k -subfactors, exponentially more than for the conventional string model.

Secondly, the conversion itself is less straightforward for unconventional string models. To illustrate this, we again consider Samala sibilant harmony (§2.3). For conventional string models, we must simply check whether some k -factor $f \in \Sigma^k$ is in G^- to determine if it is added to G^+ : if $[s\dots f] \in G^-$, for example, then $[s\dots f] \notin G^+$. For unconventional string models, however, this is not sufficient. Returning to Figure 4, if subfactor (c) is in G^- , then we should not include (b) in the corresponding G^+ , even though $b \notin G^-$, since $c \sqsubseteq b$. Likewise, if $b \in G^-$, then $c \notin G^+$. Thus, to determine whether some k -subfactor f should be added to G^+ for an unconventional string model, we must check not only if $f \in G^-$, but also if $(\exists g \in G^-)[f \sqsubseteq g \vee g \sqsubseteq f]$. Hence, both the number of possible k -subfactors and the method of conversion indicate that interdefinability is prohibitively costly for unconventional string models, further motivating the learning of positive grammars directly from these models.

5 The Learning Problem

[Heinz \(2010b\)](#) and [Heinz et al. \(2012\)](#) demonstrate that positive grammars like those in Definition 7 are learnable in the limit from positive evidence in the sense of [Gold \(1967\)](#), as well as PAC-learnable ([Valiant, 2013](#)) in some cases. In this work, G^+ is defined as the collection of all k -factors in the data sample, and a word w is in $L(G^+)$ if and only if all of its k -factors are in G^+ . Because [Heinz \(2010b\)](#) and [Heinz et al. \(2012\)](#) are not working in the model-theoretic framework, it is sufficient to check that the k -factors are in the grammar, rather than superfactors of elements in the grammar.

[Chandlee et al. \(2019\)](#) learn negative grammars from positive evidence with unconventional string models. Rather than convergence to a *correct* grammar in the limit, [Chandlee et al.](#) define the learning problem in terms of returning an *adequate* grammar given a finite positive sample, in the sense of [De Raedt \(2008\)](#). We adapt [Chandlee et al.](#)'s definition of the learning problem to apply to both negative and positive grammars as follows:

Definition 9 (The Learning Problem). Fix Σ , model M , positive integer k , and polarity p . For any language $L \in \mathcal{L}^p(M, k)$ and for any finite sample $D \subseteq L$, return a grammar G^p such that:

1. G^p is **consistent**, that is, $D \subseteq L(G^p)$.
2. $L(G^p)$ is a **smallest language** in $\mathcal{L}^p(M, k)$ which covers D , so that for all $L \in \mathcal{L}^p(M, k)$ where $D \subseteq L$, we have $L(G^p) \subseteq L$.
3. G^p includes R-structures S that are restrictions of R-structures S' in other grammars G' that also satisfy (1) and (2). That is, for all G' satisfying (1) and (2) and for all $S' \in G'$, there exists some $S \in G^p$ such that $S \sqsubseteq S'$.

The first criterion is self-explanatory: we want to at least cover the training data. Following [Chandlee et al.](#), the second criterion is motivated by [Angluin \(1980\)](#)'s analysis of identification in the limit. The third criterion requires us to learn the most *general* subfactors: for positive grammars, this means the subfactors that most generally encompass the allowed maxfactors, while for negative grammars, it means the most general constraints. In the case of Samala sibilant harmony (§2.3), for example, if we see $[j\dots f]$ and $[3\dots 3]$, but not $[s\dots f]$, $[z\dots 3]$, $[s\dots 3]$, or $[z\dots f]$, then we would add to our positive grammar $[-\text{ANT}, +\text{STR}][-\text{ANT}, +\text{STR}]$, rather than the

Algorithm 1 A generalized bottom-up learning algorithm for positive and negative grammars

Require: positive sample D , size k , and polarity p

```

1:  $Q \leftarrow \{s_0\}$ 
2:  $G \leftarrow \emptyset$ 
3:  $V \leftarrow \emptyset$ 
4: while  $|Q| > 0$  do
5:    $s \leftarrow Q.dequeue()$ 
6:    $V \leftarrow V \cup \{s\}$ 
7:   if  $((p = -) \wedge (\exists s' \in \text{EXT}_k(s), \exists x \in D)[s' \leq x]) \vee$ 
8:  $(p = +) \wedge (\exists s' \in \text{EXT}_k(s))[\forall x \in D : s' \not\leq x]$  then
9:      $S \leftarrow \text{NextSupFact}(s)$ 
10:     $S' \leftarrow \{s \in S \mid (\nexists g \in G)[g \sqsubseteq s] \wedge s \notin V\}$ 
11:     $Q.enqueue(S')$ 
12:  else
13:     $G \leftarrow G \cup \{s\}$ 
14:  end if
15: end while
16: return  $G$ 

```

two more specified factors. From the same data, we would add to our negative grammar $[+ANT, +STR][-ANT, +STR]$, rather than the four more specified factors. The third criterion is specific to unconventional string models: as discussed in §3, maxfactors and subfactors are equivalent for conventional string models, so there is no parallel notion of generality when learning over them.

6 A Generalized Learning Algorithm

The learning algorithm we present is based closely on that of Chandlee et al. (2019). Since our goal is to add the most *general* subfactors to our grammar, the algorithm is **bottom-up** in the sense of De Raedt (2008): we begin with the most general subfactors and traverse upwards in the partial order during learning. Indeed, once a subfactor has been identified as an element of the grammar, none of its superfactors need to be considered: all of them will be banned in the case of a negative grammar or allowed in the case of a positive grammar.

A bottom-up learner that learns both positive and negative grammars is given in Algorithm 1. As input, this algorithm takes a positive data sample D , an integer k corresponding to the size of the subfactors, and a polarity p indicating whether to learn a positive or negative grammar. As in Chandlee et al. (2019), Algorithm 1 makes use of a queue Q , which initially contains just the empty structure of length k , s_0 (Line 1). The algorithm also initializes two empty sets: G (Line 2), the grammar to be returned, and V (Line 3), the set of subfactors that have already been visited. Algorithm 1 considers the subfactors in Q one at a time in first-in-first-out order, and as each subfactor s is considered, it is

added to the set of visited subfactors V (Line 6). Depending on the polarity p of the grammar to be learned, we condition as follows for a given s (Line 7): For a negative grammar, we check whether *any* of the possible extensions of s is a k -maxfactor of some element in D . For a positive grammar, we check whether *any* of the possible extensions of s is *not* a k -maxfactor of any element in D . The extensions of s are defined as follows:

$$\text{EXT}_k(s) = \{A \in \text{SFAC}_k(M, \Sigma^*) \mid s \sqsubseteq A \wedge (\nexists A')[|A'| = k \wedge A \sqsubseteq A']\} \quad (18)$$

In other words, the extensions of s are all k -maxfactors that are superfactors of s . For example, if we have $s = [+ANT]$ and the only two features available are $\pm ANT$ and $\pm STR$, then we have $\text{EXT}_k(s) = \{[+ANT, -STR], [+ANT, +STR]\}$.

Given Definitions 6 and 7, if the conditions on Line 7 are not satisfied, we add s to G (Line 13). If either of the conditions are satisfied, however, we must consider more specified subfactors than s . For a negative grammar, this is because the potential constraint s is violated by some $w \in D$ and thus cannot be added to G . For a positive grammar, this is because at least one k -maxfactor licensed by s is unattested in D , and thus s cannot be added to G . We extract the more specific superfactors of s by calling $\text{NextSupFact}(s)$ (Line 9) where $\text{NextSupFact}()$ is defined as follows:

$$\text{NextSupFact}(s) = \{A \in \text{SFAC}_k(M, \Sigma^*) \mid s \sqsubseteq A \wedge (\nexists A')[s \sqsubseteq A' \sqsubseteq A]\} \quad (19)$$

Intuitively, $\text{NextSupFact}()$ returns the *least* superfactors for s . The set S of superfactors is then filtered (Line 10) to contain only those that have not been previously visited and contain no element of G as a subfactor. This is because if there is some $g \in G$ such that $g \sqsubseteq s$, then for any word w for which $s \sqsubseteq w$, we have $g \sqsubseteq s \sqsubseteq w$ and thus $g \sqsubseteq w$, and by Definitions 6 and 7, s will not add any new information to the grammar. The structures that pass this filter are then added to Q .

Note that Algorithm 1 is nearly identical to the algorithm of Chandlee et al. except that Line 7 conditions for both positive and negative grammars, and we consider subfactors of *exactly* size k rather than bounded in size by k . As discussed in §4.1, it is the latter modification that allows us to learn both positive and negative grammars in the same way. We now demonstrate that Algorithm 1 meets the criteria outlined in Definition 9.

Theorem 2. For any $p \in \{+, -\}$, positive integer k , any $L \in \mathcal{L}^p(M, k)$ and any finite set $D \subseteq L$ provided to Algorithm 1, it returns a grammar G^p satisfying Definition 9.

Proof. (**Condition 1**) Assume towards contradiction that there exists some $w \in D, w \notin L(G^p)$.

If $p = +$, then by Definition 7, there is some $x \in \text{MFAC}_k(M, w)$ such that $(\forall y \in \text{SFAC}_k(x))[y \notin G^p]$. By Algorithm 1, this means that for all $y \in \text{SFAC}_k(x)$ there is some $z \in \text{EXT}_k(y)$ such that $(\forall w' \in D)[z \not\leq w']$. However, $x \in \text{SFAC}_k(x)$ by Definition 4, and since $x \in \text{MFAC}_k(M, w)$, we have $\text{EXT}_k(x) = \{x\}$. Thus, if $x \notin G^+$, then $(\forall w' \in D)[x \not\leq w']$, so $w \notin D$, a contradiction.

If $p = -$, then by Definition 6, there is some $x \in \text{MFAC}_k(M, w)$ such that $(\exists y \in \text{SFAC}_k(x))[y \in G^p]$. Since $y \in \text{SFAC}_k(x)$ and $x \in \text{MFAC}_k(M, w)$, we have $x \in \text{EXT}_k(y)$. By Algorithm 1, $y \in G^-$ means that $(\forall y' \in \text{EXT}_k(y), \forall w' \in D)[y' \not\leq w']$, but since $x \in \text{EXT}_k(y)$, this means that $(\forall w' \in D)[x \not\leq w']$, and $w \notin D$, a contradiction.

(**Condition 2**) Consider any $L' \in \mathcal{L}^p(M, k)$ with $D \subseteq L'$ and any $w \in L(G^p)$. Since $w \in L(G^p)$, we have $\text{SFAC}_k(M, w) \subseteq \text{SFAC}_k(M, D)$ and since $D \subseteq L'$, we have $\text{SFAC}_k(M, D) \subseteq \text{SFAC}_k(M, L')$. Thus, $\text{SFAC}_k(M, w) \subseteq \text{SFAC}_k(M, L')$, and $w \in L'$. As such, $(\forall w \in L(G^p))[w \in L']$, and $L(G^p) \subseteq L'$.

(**Condition 3**) Assume towards contradiction that there is some G^p learned by Algorithm 1 such that for some $s \in G^p, \exists s' \sqsubseteq s$ that should be included in G^p : either $p = +$ and $(\forall x \in \text{EXT}_k(s'))[\exists w \in D, x \leq w]$, or $p = -$ and $(\forall x \in \text{EXT}_k(s'))[\not\exists w \in D, x \leq w]$. Since $s' \sqsubseteq s$, s' will be added to Q before s is generated by $\text{NextSupFact}()$ under Algorithm 1, and since Q is a first-in-first-out queue, s' will be removed from Q for consideration before s is generated. Since we have either $p = +$ and $(\forall x \in \text{EXT}_k(s'))[\exists w \in D, x \leq w]$, or $p = -$ and $(\forall x \in \text{EXT}_k(s'))[\not\exists w \in D, x \leq w]$, s' will be added to G by Line 7. Then, when s is generated by $\text{NextSupFact}()$, it will not pass the filter in Line 10, since $s' \sqsubseteq s$ and $s' \in G^p$. As such, s is never added to G^p , and $s \notin G^p$, a contradiction. \square

7 Example: Samala Sibilant Harmony

We illustrate our learning algorithm by applying it to a toy example based on Samala sibilant harmony (§2.3; Hansson, 2010). For simplicity, we

use only two features: $\pm\text{ANT}$ and $\pm\text{VOI}$; the former is necessary to define the phonotactic restriction and the latter is not. We define the size of the subsequences to be $k = 2$, and assume that all licit subsequences are attested in D (c.f. Heinz, 2010a). The partially-ordered structure of the hypothesis space is shown in Figure 6, with lines indicating subfactor-superfactor relations (see Chandlee et al. 2019 for further discussion).

Following Line 1, we initialize Q to contain only the empty 2-subfactor $[\]$, shown at the bottom of Figure 6. Learning begins by dequeuing and considering $[\]$ (Lines 5-6). If we are learning a negative grammar, we check whether there is any element in $\text{EXT}_k([\])$ which is a 2-maxfactor of some $x \in D$. By definition, $\text{EXT}_k([\])$ will contain all fully-specified 2-factors that are superfactors of $[\]$. This means, for example, that $[+\text{VOI}, +\text{ANT}][+\text{VOI}, +\text{ANT}] \in \text{EXT}_k([\])$, and this corresponds to the licit subsequence $[z\dots z]$ which is attested in D . If we are learning a positive grammar, we check whether any element in $\text{EXT}_k([\])$ is *not* a 2-maxfactor of any $x \in D$. We have, for example, $[+\text{VOI}, +\text{ANT}][+\text{VOI}, -\text{ANT}] \in \text{EXT}_k([\])$, and this corresponds to the illicit subsequence $[z\dots z]$ which is not attested in D . As such, the condition on Line 7 is satisfied for either polarity.

Following Line 9, we then extract the *least* superfactors of $[\]$; these are shown in the level above $[\]$ in Figure 6. Since none of these subfactors have been seen and G is empty, they are all added to Q (Lines 10-11). However, as each is dequeued and considered, it still satisfies the criteria in Line 7: any subfactor with only $\pm\text{ANT}$ specified in a single position will have both licit and illicit maxfactors in its extension (e.g., $[+\text{ANT}][\] \sqsubseteq [+\text{ANT}][+\text{ANT}]$ means that Line 7 will be satisfied for negative grammars, but $[+\text{ANT}][\] \sqsubseteq [+\text{ANT}][-\text{ANT}]$ means that it will also be satisfied for positive grammars). Similarly, any subfactor with only $\pm\text{VOI}$ specified will have both licit and illicit maxfactors in its extension. As such, specification of the subfactors under consideration will be increased once more, corresponding to the third level in Figure 6.

It is here that we are able to add subfactors to G . When $[+\text{ANT}][+\text{ANT}]$ is dequeued for the positive grammar, every factor in $\text{EXT}_k([+\text{ANT}][+\text{ANT}])$ (namely $[s\dots s]$, $[z\dots z]$, $[s\dots z]$, and $[z\dots s]$) is attested, so Line 7 is not satisfied, and $[+\text{ANT}][+\text{ANT}]$ is added to G (Line 13). Similarly, when $[+\text{ANT}][-\text{ANT}]$ is dequeued for the negative grammar, no factor in $\text{EXT}_k([+\text{ANT}][-\text{ANT}])$ (i.e., none of $[s\dots f]$,

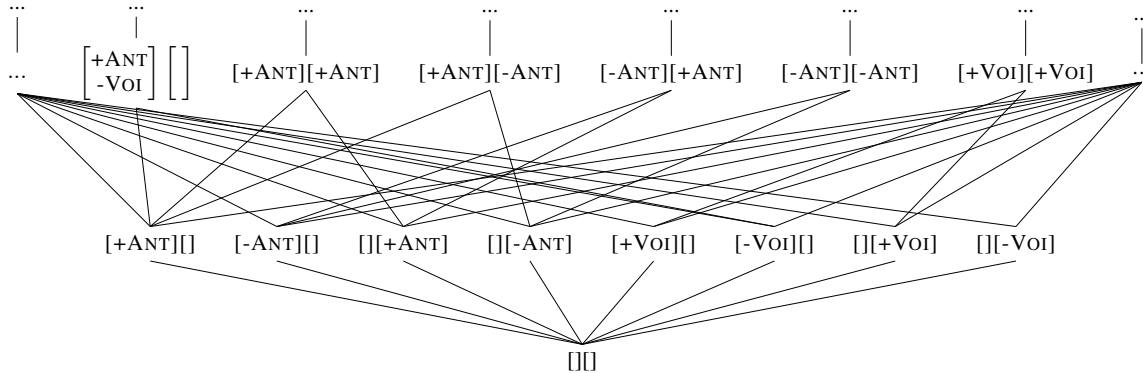


Figure 6: A partial illustration of the hypothesis space for learning Samala sibilant harmony with $k = 2$.

[s...ʒ], [z...ʒ] or [s...ʒ]) is attested, so [+ANT][-ANT] is added to G (Line 13). We may later de-queue [+ANT][+ANT, +VOI] in the positive case, but will not consider it (Line 10): [+ANT][+ANT] being allowed entails [+ANT][+ANT, +VOI] being allowed, so there is no reason to consider [+ANT][+ANT, +VOI] separately. Similarly, [+ANT][-ANT, +VOI] will not be considered in the negative case since [+ANT][-ANT] being banned entails [+ANT][-ANT, +VOI] being banned.

8 Discussion

The polarity of the grammar has several implications that warrant future exploration. In implementation, it is often necessary to terminate the search defined in Algorithm 1 before reaching the most specific k -maxfactors, but the implications of this termination differ based on the polarity of the grammar. Consider some positive data sample D , and let G^p be the grammar that will be learned by Algorithm 1 from D if the search space is traversed in its entirety. Let $G^p(t)$ be the intermediate grammar at some time t . It is easy to see that $G^p(t) \subseteq G^p$, since at any time t , elements in Q — as well as their superfactors — have not yet been considered.

However, the implications of $G^p(t) \subseteq G^p$ differ depending on the value of p . Specifically, $L(G^+(t)) \subseteq L(G^+)$ but $L(G^-) \subseteq L(G^-(t))$, since the additional elements in $G^p \setminus G^p(t)$ will either be interpreted as additional constraints (for negative p) or additional permitted elements (for positive p). Recall from §4.1 that $L^+(\emptyset) = \emptyset$ and $L^-(\emptyset) = \Sigma^*$, and from Algorithm 1 that G^p is initialized to \emptyset . This, in conjunction with the subset relations above, entails that Algorithm 1 consistently expands $L(G^+)$ during learning of a positive grammar by adding more allowed subfactors to G^+ ,

while it consistently shrinks $L(G^-)$ during learning of a negative grammar by adding more banned subfactors to G^- . Future work should investigate how these differing predictions map onto developmental findings. While some findings have suggested an initial stage of conservatism in child productions (Fikkert, 1994; Levelt et al., 2000; Rose, 2000, i.a.), there is also evidence for early generalization in perception (Cristia and Peperkamp, 2012; Hallé and Cristia, 2012; Bernard and Onishi, 2023, i.a.), particularly based on phonological features and syllable position. Do children begin by positing that *anything* is allowed and later backtrack, or do they begin by positing that *nothing* is allowed, and only add items to their grammar once they have been observed in the input?

9 Conclusion

In this paper, we showed that if we fix the size k of subfactors in the grammar, then the algorithm of Chandlee et al. (2019) can be straightforwardly extended to learn both positive and negative grammars over unconventional string models in a unified way. The enriched representations provided by unconventional string models allow us to provably find the most general subfactors that are allowed or banned in a given language by conducting a bottom-up search of the partial ordering of k -subfactors.

Acknowledgements

I am grateful to Jeff Heinz, Thomas Graf, Jon Rawski, Logan Swanson, and the SCiL reviewers for their feedback. This work was supported by the Institute for Advanced Computational Science Graduate Research Fellowship and the National Science Foundation Graduate Research Fellowship Program under NSF Grant No. 2234683.

References

- Dana Angluin. 1980. Inductive inference of formal languages from positive data. *Information and Control*, 45(2):117–135.
- Caleb Belth. 2023. *Towards an Algorithmic Account of Phonological Rules and Representations*. Ph.D. thesis, University of Michigan.
- Caleb Belth, Sarah Payne, Deniz Beser, Jordan Kodner, and Charles Yang. 2021. The greedy and recursive search for morphological productivity. *Proceedings of the 43rd annual meeting of the Cognitive Science Society*, 43:2869–2875.
- Amélie Bernard and Kristine H Onishi. 2023. Novel phonotactic learning by children and infants: Generalizing syllable-position but not co-occurrence regularities. *Journal of Experimental Child Psychology*, 225:105493.
- J Richard Büchi. 1960. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6).
- Jane Chandlee, Remi Eyraud, Jeffrey Heinz, Adam Jardine, and Jonathan Rawski. 2019. [Learning with partially ordered representations](#). In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 91–101. Toronto, Canada. Association for Computational Linguistics.
- Alejandrina Cristia and Sharon Peperkamp. 2012. Generalizing without encoding specifics: Infants infer phonotactic patterns on sound classes. In *Proceedings of the 36th Annual Boston University Conference on Language Development (BUCLD 36)*, pages 126–138.
- Luc De Raedt. 2008. *Logical and relational learning*. Springer Science & Business Media.
- Herbert B Enderton. 2001. *A mathematical introduction to logic*. Elsevier.
- Paula Fikkert. 1994. *On the acquisition of prosodic structure*. ICG Printing.
- E Mark Gold. 1967. Language identification in the limit. *Information and Control*, 10(5):447–474.
- Pierre Hallé and Alejandrina Cristia. 2012. Global and detailed speech representations in early language acquisition. In *Speech planning and dynamics*, pages 11–38. Peter Lang.
- Gunnar Ólafur Hansson. 2010. *Consonant harmony: Long-distance interactions in phonology*, volume 145. University of California Press.
- Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, 39(3):379–440.
- Jeffrey Heinz. 2010a. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.
- Jeffrey Heinz. 2010b. [String extension learning](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906. Uppsala, Sweden. Association for Computational Linguistics.
- Jeffrey Heinz. 2016. Computational theories of learning and developmental psycholinguistics. In Jeffrey Lidz, William Snyder, and Joe Pater, editors, *The Oxford Handbook of Developmental Linguistics*, pages 633–663. Oxford University Press, Oxford, UK.
- Jeffrey Heinz. 2018. The computational nature of phonological generalizations. *Phonological typology, phonetics and phonology*, pages 126–195.
- Jeffrey Heinz, Anna Kasprzik, and Timo Kötzing. 2012. [Learning in the limit with lattice-structured hypothesis spaces](#). *Theoretical Computer Science*, 457:111–127.
- Dakotah Lambert, Jonathan Rawski, and Jeffrey Heinz. 2021. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling*, 9.
- Clara C Levelt, Niels O Schiller, and Willem J Levelt. 2000. The acquisition of syllable types. *Language acquisition*, 8(3):237–264.
- Daoxin Li and Kathryn D Schuler. 2023. Acquiring recursive structures through distributional learning. *Language Acquisition*, pages 1–14.
- Leonid Libkin. 2004. *Elements of finite model theory*, volume 41. Springer.
- Gary F Marcus, Steven Pinker, Michael Ullman, Michelle Hollander, T John Rosen, Fei Xu, and Harald Clahsen. 1992. Overregularization in language acquisition. *Monographs of the society for research in child development*, pages i–178.
- Robert McNaughton and Seymour A Papert. 1971. *Counter-Free Automata (MIT research monograph no. 65)*. The MIT Press.
- Sarah Payne. 2023. Marginal sequences are licit but unproductive. Poster presented at the 2023 Annual Meeting of Phonology.
- Alan Prince and Paul Smolensky. 1993. *Optimality Theory: Constraint interaction in generative grammar*. John Wiley & Sons.
- Jonathan Rawski. 2021. *Structure and Learning in Natural Language*. Ph.D. thesis, State University of New York at Stony Brook.
- James Rogers and Jeffrey Heinz. 2014. Model theoretic phonology. In *Workshop slides in the 26th European Summer School in Logic, Language and Information*.
- 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 *The Mathematics of Language: 10th and*

11th Biennial Conference, Revised Selected Papers, pages 255–265. Springer.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal Grammar: 17th and 18th International Conferences, Revised Selected Papers*, pages 90–108. Springer.

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

Yvan Rose. 2000. *Headedness and prosodic licensing in the L1 acquisition of phonology*. Ph.D. thesis, McGill University.

Kristina Strother-Garcia, Jeffrey Heinz, and Hyun Jin Hwangbo. 2016. Using model theory for grammatical inference: a case study from phonology. In *Proceedings of The 13th International Conference on Grammatical Inference*, pages 66–78.

Leslie Valiant. 2013. *Probably approximately correct: nature's algorithms for learning and prospering in a complex world*. Basic Books.

Mai H Vu, Ashkan Zehfroosh, Kristina Strother-Garcia, Michael Sebok, Jeffrey Heinz, and Herbert G Tanner. 2018. Statistical relational learning with unconventional string models. *Frontiers in Robotics and AI*, 5:76.

Charles Yang. 2016. *The price of linguistic productivity: How children learn to break the rules of language*. MIT press.

A Proof of Theorem 1

Proof. To construct G_2^- from G_1^- , for each $g \in G_1^-$, if $|g| = k$, then add g directly to G_2^- . If $|g| < k$, then add to G_2^- all superfactors of g of length k given by $\{f \mid g \sqsubseteq f, |f| = k\}$. Assume towards contradiction that $L_{17}(G_1^-) \neq L_{11}(G_2^-)$. This means that either $(\exists w)[w \in L_{17}(G_1^-), w \notin L_{11}(G_2^-)]$ or $(\exists w)[w \in L_{11}(G_2^-), w \notin L_{17}(G_1^-)]$.

(Case 1) $(\exists w)[w \in L_{17}(G_1^-), w \notin L_{11}(G_2^-)]$: By Equation (11), $w \notin L_{11}(G_2^-)$ means that:

$$(\exists S \in \text{MFAC}_k(M, w))[\text{SFAC}_k(S) \cap G_2^- \neq \emptyset]$$

or equivalently that:

$$\left(\exists f \in \bigcup_{S \in \text{MFAC}_k(M, w)} \text{SFAC}_k(S) \right) [f \in G_2^-]$$

By Lemma 2 this means that:

$$[\exists f \in \text{SFAC}_k(M, w)](f \in G_2^-)$$

Given our construction of G_2^- , it is either the case that $f \in G_1^-$ or that $[\exists g \in G_1^-](g \sqsubseteq f)$. In the first case, $f \in \text{SFAC}_k(M, w) \subseteq \text{SFAC}_{\leq k}(M, w) \Rightarrow f \in \text{SFAC}_{\leq k}(M, w)$, but if $f \in \text{SFAC}_{\leq k}(M, w)$ and $f \in G_1^-$, then $\text{SFAC}_{\leq k}(M, w) \cap G_1^- \neq \emptyset$ and $w \notin L_{17}(G_1^-)$, a contradiction. In the second case, $g \sqsubseteq f \in \text{SFAC}_k(M, w) \Rightarrow g \in \text{SFAC}_{\leq k}(M, w)$, but if $g \in \text{SFAC}_{\leq k}(M, w)$ and $g \in G_1^-$, then $\text{SFAC}_{\leq k}(M, w) \cap G_1^- \neq \emptyset$ and $w \notin L_{17}(G_1^-)$, a contradiction.

(Case 2) $(\exists w)[w \in L_{11}(G_2^-), w \notin L_{17}(G_1^-)]$: By Equation 17, $w \notin L_{17}(G_1^-)$ means that

$$[\exists g \in \text{SFAC}_{\leq k}(M, w)](g \in G_1^-)$$

Given our construction of G_2^- , it is either the case that $|g| = k$ and $g \in G_2^-$ or that $[\forall f \mid g \sqsubseteq f, |f| = k](f \in G_2^-)$. In the first case, since $|g| = k$, $g \in \text{SFAC}_k(M, w)$ and thus:

$$g \in \bigcup_{S \in \text{MFAC}_k(M, w)} \text{SFAC}_k(S)$$

by Lemma 2. But this, in conjunction with $g \in G_2^-$, means that:

$$\left(\bigcup_{S \in \text{MFAC}_k(M, w)} \text{SFAC}_k(S) \right) \cap G_2^- \neq \emptyset$$

and $w \notin L_{11}(G_2^-)$, a contradiction. In the second case, $g \in \text{SFAC}_{\leq k}(M, w) \Rightarrow [\exists g' \in \text{SFAC}_k(M, w)](g \sqsubseteq g')$, but since $\{f \mid g \sqsubseteq f, |f| = k\} \subseteq G_2^-$, it must be the case that $g' \in G_2^-$. Since $g' \in \text{SFAC}_k(M, w)$ and $g' \in G_2^-$, by Lemma 2:

$$\left(\bigcup_{S \in \text{MFAC}_k(M, w)} \text{SFAC}_k(S) \right) \cap G_2^- \neq \emptyset$$

and $w \notin L_{11}(G_2^-)$, a contradiction. \square

Pragmatics-utilizing distributional learner (PUDL) without deterministic hypothesis space

Boram Kim

Department of Linguistics
University of California, Los Angeles
bkim21@ucla.edu

Joonsuk Kang

Department of Statistics
University of Chicago
joonsukkang@uchicago.edu

Abstract

We introduce Pragmatics-Utilizing Distributional Learner (PUDL) to simulate verb transitivity learning in 15-month-old English learners. The model incorporates pragmatic reasoning about question-answer relations in neutral wh-questions. Our proposal outlines a developmental trajectory that features a temporary overregularization stage where learners generalize all verbs into one category, due to difficulty in distinguishing Prepositional Phrases from Noun Phrase objects. The results demonstrate the effectiveness of a pure distributional model enhanced by pragmatic knowledge in addressing learning challenges posed by noisy input.

1 Introduction

Learning how verbs behave in terms of taking direct objects proves to be a challenging task for learners. The complexity of verb transitivity learning arises from messy data that learners encounter, as illustrated in (1-2).

- (1) *transitive*
 - a. Alex threw the truck.
 - b. What did Alex throw?
 - c. *Alex threw.
- (2) *intransitive*
 - a. I waited.
 - b. I waited for Alex.

In an ideal setting for learning verb transitivity, learners would be exposed solely to examples featuring transitive verbs with a direct object (1a) and intransitive verbs without any adverbials (2a). With this transparent input, they could seamlessly make inferences about the transitivity patterns of verbs. However, reality often deviates from this ideal, exposing learners to less-than-optimal examples. In the surface string of (1b), the transitive verb

‘throw’ does not take any direct object due to the English rule of question formation: a direct object moves to the beginning in object wh-questions. A learner who hasn’t yet acquired this non-local wh-dependency might be misled to infer from (1b) that ‘throw’ does not always take a direct object. In contrast, ‘wait’ is an intransitive verb that does not take a direct object, as demonstrated by the contrast between (1c) and (2a). Confusingly for learners though, prepositional phrases (PP) such as ‘for Alex’ in (2b) often occur with the intransitive verb ‘wait.’ Novice learners, who have yet to acquire the distinction between the PP ‘for Alex’ in (2b) and the NP ‘the truck’ in (1a), might incorrectly infer from utterances like (2b) that ‘wait’ is a transitive verb. The abundance of utterances like (1b) and (2b) in learners’ input prompts a critical question: How do learners, faced with such misleading input, eventually arrive at accurate generalizations that transitive verbs like ‘throw’ consistently require a direct object, while intransitive verbs like ‘wait’ do not take a direct object?

Assuming grammatical knowledge of learners around 15 months old, we hypothesize that (i) pragmatic reasoning is what enables them to realize questions like (1b) do not serve as evidence for the intransitive nature of transitive verbs, but (ii) due to the failure to distinguish NP arguments (e.g., ‘the truck’ in (1a)) from PP adjuncts (e.g., ‘for Alex’ in (2b)) at the proposed developmental stage, they undergo a temporary overregularization stage where they perceive all verbs as transitives, on their way to the final destination, i.e., adult grammar. Our assumption about the developmental timeline is directly motivated by experimental results. Behavioral studies show that 15-month-olds behave as if they comprehend wh-dependency in (1b) (Gagliardi et al., 2016; Perkins and Lidz, 2021).¹ On the other hand, it has been experimen-

¹In this regard, our claim about pragmatic reasoning can

tally shown that children as young as 19 months old incorrectly interpret PPs (she’s wiping *with the tig*) as denoting a patient, a thematic role typically expressed by a direct object (she’s wiping *the tig*) (Lidz et al., 2017). In other words, the learners we assume have overcome the learning problem that arises in the transitive domain (1), but not in the intransitive domain (2).

To model our target learner, English-learning 15-month-olds capable of pragmatic reasoning, but not PP vs. NP resolution, we propose Pragmatics-Utilizing Distributional Learner (PUDL). Using Bayesian Information Criterion (BIC), we show that the PUDL goes through an overregularization phase where it prefers all verbs to be transitives. Compared to pure distributional learner (DL), which is not pragmatically informed, the PUDL’s performance is farther from true knowledge about verb transitivity patterns, when asked to cluster verbs into three groups (transitive, alternating, intransitive). Still, pragmatic reasoning is hypothesized to be crucial to grappling with the misleading data of (1b) kind in the transitive domain; once learners become question-savvy, they are not tricked anymore by (1b). The resulting overregularization inference that every verb takes a direct object is inevitable given the messy nature of data they receive in the intransitive domain; 15-month-olds frequently hear utterances like (2b), while perceiving PPs incorrectly as NP objects.

The proposal is consistent with the idea that regularization, in general, plays a pivotal role in both first and second language acquisition (e.g., Hudson Kam and Newport 2005; Austin et al. 2022). Furthermore, we show that a pure distributional learner, as opposed to a learner with additional inductive bias, such as filtering (Perkins et al., 2022), is just as promising to tackle the puzzle in verb transitivity learning, although a full comparison with the PUDL augmented by the PP vs. NP resolution is left for future research.

2 Pragmatics-utilizing distributional learner (PUDL /pudəl/)

We propose Pragmatics-Utilizing Distributional Learner (PUDL), a pure distributional model that sidesteps deterministic hypotheses as part of its inductive bias but is bolstered by pragmatic knowledge. The base model we start with is distribu-

be understood as an attempt to answer how such knowledge of wh-dependency arises in young learners.

tional learner (DL). In the proposed model, verb categories do not have a fixed direct object (DO) probability; instead, they have probability distributions over the interval [0,1]. Intuitively, our learner operates with confidence in the received data, compared to alternative learners that filter out some proportion of data for successful learning. Without knowing that the input is noisy, the PUDL perceives every piece of data, including (1b) and (2b), as a valuable signal, as is reasonable to be assumed for learners as young as 15 months old who have no clue about deterministic verb transitivity. We assume that all they are sensitive to is the distributional patterns of verb transitivity.

The central challenge for our base model concerns a transition to acquiring correct deterministic knowledge without relying on a predefined deterministic hypothesis space. We propose that pragmatic understanding of discourse context plays a crucial role in addressing this issue for transitive verbs. Specifically, recognizing that (1b) functions as a neutral question that seeks information facilitates learners’ transitivity acquisition. For instance, let’s assume, for illustrative purposes, that the verb ‘throw’ occurs in the form of (1a) 80% of the time in the input, while 20 % of the time, it takes the form of (1b). Based on the observations from the input, a learner would form immature knowledge that ‘throw’ occurs with a direct object only 80% of the time. Once pragmatically informed, however, the learner associates the remaining 20% or so, due to (1b), with the information-seeking discourse function inherent in wh-questions. It would then cease its search for a missing direct object in interrogative sentences, recognizing that such information-seeking sentences are supposed to lack a direct object, i.e., the *answer* of the question. This nuanced yet straightforward pragmatic reasoning prompts the learner to update the initial underestimated knowledge about ‘throw.’ As a result, the learner moves closer to the correct understanding that ‘throw’ should always occur with a direct object, ideally reaching near 100%. The gap, previously attributed to ‘throw’s intrinsic property, is now ascribed to a specific discourse context of information seeking, which allows verbs to lack a direct object.

Two concerns may arise regarding (i) whether the complexity of the proposed pragmatic reasoning is appropriate for a learner as young as 15 months old, and (ii) imperfect correlation between missing direct objects and questions. First, despite the dis-

course function of questions being more complex than its declarative counterpart, two factors are hypothesized to enhance learners' capacity for the proposed pragmatic reasoning: (i-a) the prevalence of questions in child-directed speech, verifiable from corpus, and (i-b) distinctive rising intonation associated with questions. On the second point (ii) regarding the imperfect correlation, it is true that not every noise in the data takes the form of question. For example, transitive verbs used in relative clauses (3a) and in passives (3b) also lack direct objects. The noisy input of these kinds would prevent even the question-savvy learner from reaching 100%, i.e., acquiring deterministic knowledge found in adult grammar.

- (3) a. I found the truck Alex threw.
 b. The truck was thrown.

We assume that a learner at this stage, where they just start to distinguish questions from non-questions, indeed fails to attain 100% correct knowledge about verbs' transitivity property. Understanding complex constructions such as relative clauses and passives likely happens later in a child's life, whether it involves a pragmatic process or not. The upshot is that the presence of other kinds of misleading data such as (3) does not argue against the plausibility of the PUDL's learning schema and the proposed developmental trajectory.

A more serious challenge to the PUDL is that not all questions take the exact form of object wh-question in (1b). Polar questions (4a), rising declaratives (4b), and subject wh-questions (4c) do not lack direct objects even though they are questions.

- (4) a. Did you throw the truck?
 b. You threw the truck?
 c. Who threw the truck?

However, polar questions (4a) and rising declaratives (4b) involve different discourse contexts from those of wh-questions in that they are *biased*. [Bur-ing and Gunlogson \(2000\)](#) argue that positive polar questions like (4a) are not neutral; they can be felicitously asked in the presence of compelling contextual evidence. Similarly, rising declaratives, extensively studied in semantics, are biased questions, where the addressee might be asked for information, but the speaker is not neutral in their expectation (see, for example, [Farkas and Roelofsen \(2017\)](#) for formal modeling of the latter discourse behavior). Therefore, it is reasonable to assume

that a learner can distinguish the discourse function of neutral wh-questions (seeking information without any expectations; (1b)) from non-neutral polar questions or rising declaratives (4a-b), which express the speaker's bias or may not necessarily expect an information-bearing answer.

Furthermore, the questions in (4) do not pose a challenge for verb transitivity learning in the first place. While a learner at the proposed stage may not correctly parse or understand each question in (4), the data are not misleading in terms of learning verb transitivity because 'throw' has a direct object in all three questions of (4). We proceed with the assumption that subject wh-questions of the (4c) kind are not noisy and, therefore, do not influence the learner's transitivity acquisition during the assumed developmental phase. In this phase, the transitivity-learning learner grapples with transparently noisy data, such as the example given in (1b). Whenever a violation of transitivity is observed as in (1b) (modulo relative clauses and passives), the PUDL associates the utterance with its unique discourse context, that is, seeking information by asking a question, and treats it as occurring with a direct object, even if the utterance (1b) lacks a direct object on the surface.

3 Data

The data we utilized are several corpora of child-directed speech from CHILDES ([MacWhinney, 2000](#)), specifically Brown ([Brown, 1973](#)), Soderstrom ([Soderstrom et al., 2008](#)), Suppes ([Suppes, 1974](#)), and Valian ([Valian, 1991](#)). Regarding the selection of corpora and the specific set of verbs, we followed [Perkins et al. \(2022\)](#) for a transparent comparison (Section 6). To model verb transitivity learning, they chose the 50 most frequent action verbs, classified into transitive, alternating, and intransitive categories.

Given our goal to model a learner around 15 months old, who has not yet resolved the NP vs. PP distinction, our learner blindly treats many elements following a verb as a direct object. Crucially, sentences like (2b) are coded as having a direct object (DO), from the learner's perspective. However, we excluded particles that make up a phrasal verb or simple adverbs from being considered as a direct object. For instance, for the verb 'pick', the utterance 'I picked up' or 'Did you pick up?' is coded as occurring without a direct object, even though the verb in question is followed by something other

than punctuation.

In addition, each sentence is coded as being a question or not. We coded a sentence as a question if and only if the sentence occurs with a question mark in its transcript, which includes a lot of rising declaratives. Then, we defined pragmatics-augmented direct object (PDO) as 1 if and only if the sentence either has a DO or is a question, and 0 otherwise. The PDO coding is used as the input for the PUDL, which utilizes pragmatics, while the DO coding is used as the input for the distributional learner DL, not equipped with pragmatic knowledge.

The list of the 50 verbs with their total counts, sample DO rates, and sample PDO rates are shown in Table 1. Verbs are categorized according to their underlying true transitivity types following Perkins et al. (2022): (T)ransitive, (A)lternating, and (I)ntransitive. They are sorted by sample DO rates within each transitivity type. Transitive verbs tend to have higher sample DO rates and intransitive verbs tend to have lower sample DO rates. However, they can deviate much from the ground truth of 1 for transitive verbs and 0 for intransitive verbs. There is also a significant overlap of the sample DO rates among the three categories.

Finally, for each verb, its sample PDO rate is always higher than its sample DO rate as expected. For all the transitive verbs, the sample PDO rate is greater than 0.9, and one verb (‘feed’) attains a 100% sample PDO rate.

4 An empirical Bayes model for distributional learning

We propose an empirical Bayes (EB) model that conducts distributional learning of verb transitivity from observed DO patterns.

Model The model assumes that there are K transitivity categories $\{C_1, C_2, \dots, C_K\}$ with equal prior weights. The transitivity T_i of each verb $i \in \{1, 2, \dots, V\}$ is distributed as:

$$T_i \sim \text{Uniform}(\{C_1, C_2, \dots, C_K\}).$$

Depending on transitivity category (C_k), the verb’s true observable DO rate θ_i is drawn from an unknown Beta distribution ($\text{Beta}(\alpha_k, \beta_k)$), taking values between 0 and 1:

$$\theta_i | T_i = C_k \sim \text{Beta}(\alpha_k, \beta_k).$$

Verb	Count	DO Rate	PDO Rate
(T) feed	226	0.9690	1.0000
(T) hit	214	0.9579	0.9860
(T) bring	712	0.9424	0.9803
(T) throw	376	0.9282	0.9415
(T) fix	397	0.8992	0.9270
(T) buy	356	0.8989	0.9775
(T) hold	565	0.8690	0.9522
(T) catch	216	0.7731	0.9074
(T) wear	540	0.7241	0.9444
(A) pick	390	0.9410	0.9692
(A) drop	178	0.9157	0.9551
(A) knock	149	0.9128	0.9664
(A) touch	210	0.8857	0.9143
(A) push	348	0.8707	0.9282
(A) wash	236	0.8686	0.9576
(A) ride	243	0.8683	0.9630
(A) turn	470	0.8617	0.9277
(A) cut	318	0.8491	0.9403
(A) lose	200	0.8450	0.9000
(A) pull	383	0.8433	0.8799
(A) read	624	0.8301	0.8942
(A) leave	382	0.8246	0.8717
(A) build	307	0.8176	0.9479
(A) open	379	0.8153	0.8707
(A) bite	195	0.7949	0.9026
(A) close	212	0.7877	0.8491
(A) blow	214	0.7570	0.8738
(A) play	1424	0.7514	0.8820
(A) drink	345	0.7507	0.9420
(A) draw	401	0.7481	0.9202
(A) eat	1535	0.7036	0.8997
(A) sit	990	0.6939	0.8323
(A) move	260	0.6923	0.7846
(A) sing	347	0.6916	0.8646
(A) hang	168	0.6905	0.8690
(A) break	558	0.6900	0.7975
(A) write	593	0.6830	0.8499
(A) walk	255	0.6196	0.8078
(A) stand	300	0.5733	0.7800
(A) stick	278	0.5647	0.7626
(A) fit	211	0.5498	0.7536
(A) jump	189	0.5185	0.7354
(A) run	246	0.4837	0.7236
(A) swim	200	0.4500	0.7550
(I) wait	310	0.8452	0.8774
(I) stay	334	0.7575	0.8204
(I) sleep	419	0.4678	0.7709
(I) fall	606	0.3449	0.6188
(I) work	302	0.3377	0.5927
(I) cry	272	0.2647	0.6875

Table 1: Fifty verbs in our analysis with their total count, 89 sample DO rate, and sample PDO rate.

Lastly, we assume that the DO observations $\{X_{i,j}\}_{j=1}^{N_i}$ are independently and identically distributed as a Bernoulli distribution with the success parameter equal to θ_i :

$$X_{i,j}|\theta_i \sim \text{Bernoulli}(\theta_i).$$

The left panel of Figure 1 summarizes our model in plate notation. Note that the verb’s transitivity T_i and the verb’s true observable DO rate θ_i are latent variables that need to be estimated, while the DO observation $X_{i,j}$ are observed variables (shaded in the Figure).

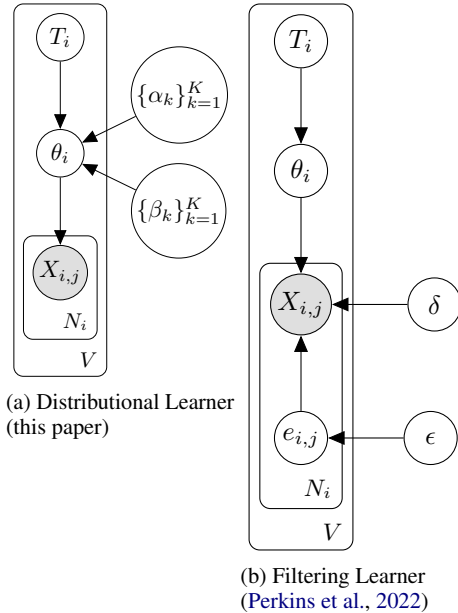


Figure 1: Models in plate notation.

EB inference We have assumed that the model hyperparameters $\{(\alpha_k, \beta_k)\}_{k=1}^K$ are unknown. We estimate these hyperparameters using EB. Specifically, we set the hyperparameters to values that maximize the marginal log-likelihood.

The EB prior estimation and posterior computation can be done efficiently by reducing our model to the class of Beta-Binomial mixture models. We combine two simple observations: the marginal distribution of θ_i is a Beta mixture if we integrate T_i out; and the sum of the N_i Bernoulli trials is distributed as a Binomial distribution, $X_{i,\cdot} := \sum_{j=1}^{N_i} X_{i,j} \sim \text{Binomial}(N_i, \theta_i)$. Therefore, the sum of DO observations $X_{i,\cdot}$ is marginally distributed as a Beta-Binomial mixture:

$$X_{i,\cdot} \sim \frac{1}{K} \sum_{k=1}^K \text{Beta-Binomial}(N_i, \alpha_k, \beta_k).$$

We use the expectation–maximization (EM) algorithm to find the hyperparameter values that maximize this likelihood.

Initialization Since the likelihood maximization problem is not a convex problem, the solution obtained via the EM algorithm might depend on the initialization. We initialize the category memberships using a hard clustering of sample DO rates, $X_{i,\cdot}/N_i$.² For example, with $K = 3$ categories, we sort verbs by their sample DO rates, and assign a hard $C_1/C_2/C_3$ membership to the verbs with sample DO rates in the lowest/middle/upper tertile, respectively. The categories C_1 , C_2 , and C_3 are interpreted as the verb categories with ‘low’, ‘middle’, and ‘high’ true observable DO rates, which would roughly correspond to the ‘intransitive’, ‘alternating’, and ‘transitive’ categories of verb transitivity for the current problem.

Inference with PDO data To make inferences using the PDO data instead of the DO data, we use the same model and algorithm. The only difference is the interpretation of the model parameters: θ_i as the verb’s true observable PDO rate and (α_k, β_k) as the parameters for the PDO distribution of the category C_k .

5 Results

We use the EB model to simulate a distributional learner (DL) that learns verb transitivity from DO data, and a pragmatics-utilizing distributional learner (PUDL) that learns verb transitivity from pragmatics-augmented DO (PDO) data, which incorporates pragmatic knowledge about questions.

5.1 Distributional Learner (DL)

To simulate a DL, we fit the EB model with three categories ($K = 3$), consistent with the underlying truth that there are three verb transitivity categories (intransitive, alternating, and transitive).

The estimated hyperparameters (α_k, β_k) for the EB Beta priors are (4.76, 3.64), (28.58, 8.60), and (33.39, 4.28) for the categories C_1 , C_2 , and C_3 ; their means are 0.57, 0.77, and 0.89. The densities of the three distributions are shown in the uppermost panel of Figure 2. Note that we do not use the

²In a hard clustering, each verb i belongs to only one category, whereas, in a soft clustering, it can belong to multiple categories. It is worth noting that the hard clustering-based initialization is an initialization strategy, not a part of the model specification, though the initialization can have a lasting impact on the final inference.

true verb transitivity labels (‘intransitive’, ‘alternating’, or ‘transitive’) in the estimation procedure.

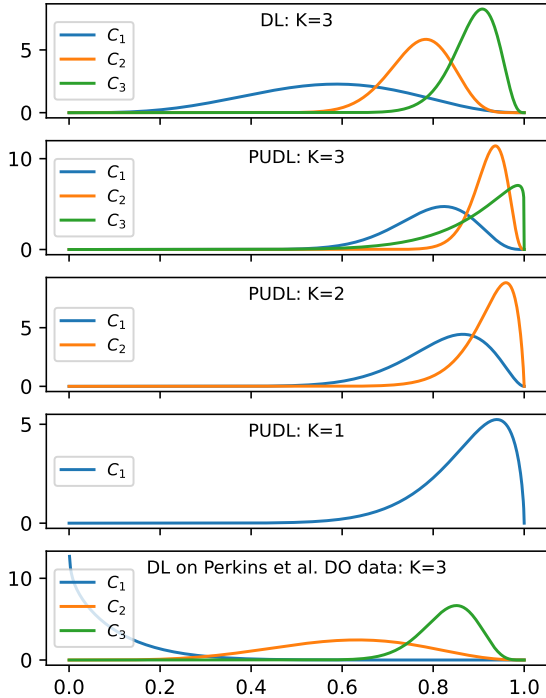


Figure 2: Empirical Bayes beta priors.

Based on the empirical Bayes beta priors, each verb’s posterior distribution over the verb categories C_1, \dots, C_K is computed. Each verb’s posterior membership in the categories is shown as a stacked bar plot in the uppermost panel of Figure 3; the posterior memberships are non-negative and sum to one. The verb labels in the x-axis are color-coded to represent the underlying true transitivity category: transitive verbs are coded red, alternating verbs are coded black, and intransitive verbs orange. The verbs are ordered first by the underlying true transitivity category, and then by the descending sample DO rate within each category.

Our EB model performs well in uncovering the underlying true transitivity category, though not perfectly. Out of the nine transitive verbs, seven verbs have the highest membership in the ‘high’ category C_3 , which is the category with the highest prior DO rates; the other two transitive verbs have the highest membership in the ‘middle’ category C_2 . On the other side, four out of the six intransitive verbs have the highest membership in the ‘low’ category C_1 . The alternating verbs have varying levels of memberships in the three categories, depending on their sample DO rates.

5.2 Pragmatics-Utilizing Distributional Learner (PUDL)

To simulate a PUDL, we fit the EB model with three categories ($K = 3$) to the PDO data. The estimated EB beta priors and the posterior memberships are shown in the second uppermost panels of Figure 2 and 3. In Figure 3, verbs within each category are reordered according to their sample PDO rates. Compared to the DL, the PUDL has verbs’ posterior memberships less separated. For example, all the fifty verbs have non-negligible memberships in the C_3 category, and the transitive verbs’ C_3 membership decreased. This change follows from the property of the PDO data: each verb’s PDO rate is always greater than or equal to its DO rate, and the verbs’ PDO rates are harder to separate into distinct clusters, since they are all shifted toward 1 (closer to 1 than the DO rates are). This property is illustrated in the estimated EB beta priors in the second panel of Figure 2, which is more overlapping than the first panel.

We find that the PUDL favors models with a smaller number of categories, based on the Bayesian Information Criterion (BIC). BIC is a criterion for model selection, which is defined as

$$\text{BIC} = -2 \log(\hat{L}) + P \log(N)$$

where \hat{L} is the maximized log-likelihood of the model, P is the number of parameters estimated by the model, and N is the sample size. A model with a smaller BIC is preferred. To strike a balance between model fit and model complexity, BIC adds a penalty to the number of parameters, as models with a larger number of parameters are more flexible to guarantee a higher maximized log-likelihood.

K	BIC	$-2 \log(\hat{L})$	$P \log(N)$
✓ 1	478.5759	470.7519	7.8240
2	486.3513	470.7032	15.6481
3	493.5469	470.0747	23.4721

Table 2: Bayesian Information Criterion for PUDL.

In our case, the sample size N is 50 and the number of parameters P is $2K$ from the size of the set $\{(\alpha_k, \beta_k)\}_{k=1}^K$. The PUDL with $K = 3$ has BIC 493.55, BIC 486.35 with $K = 2$, and 478.58 with $K = 1$ (see Table 2). Therefore, the PUDL with $K = 1$ is the most preferred, and the PUDL with $K = 3$ is the least preferred, among the three models. The estimated EB prior for the PUDL with

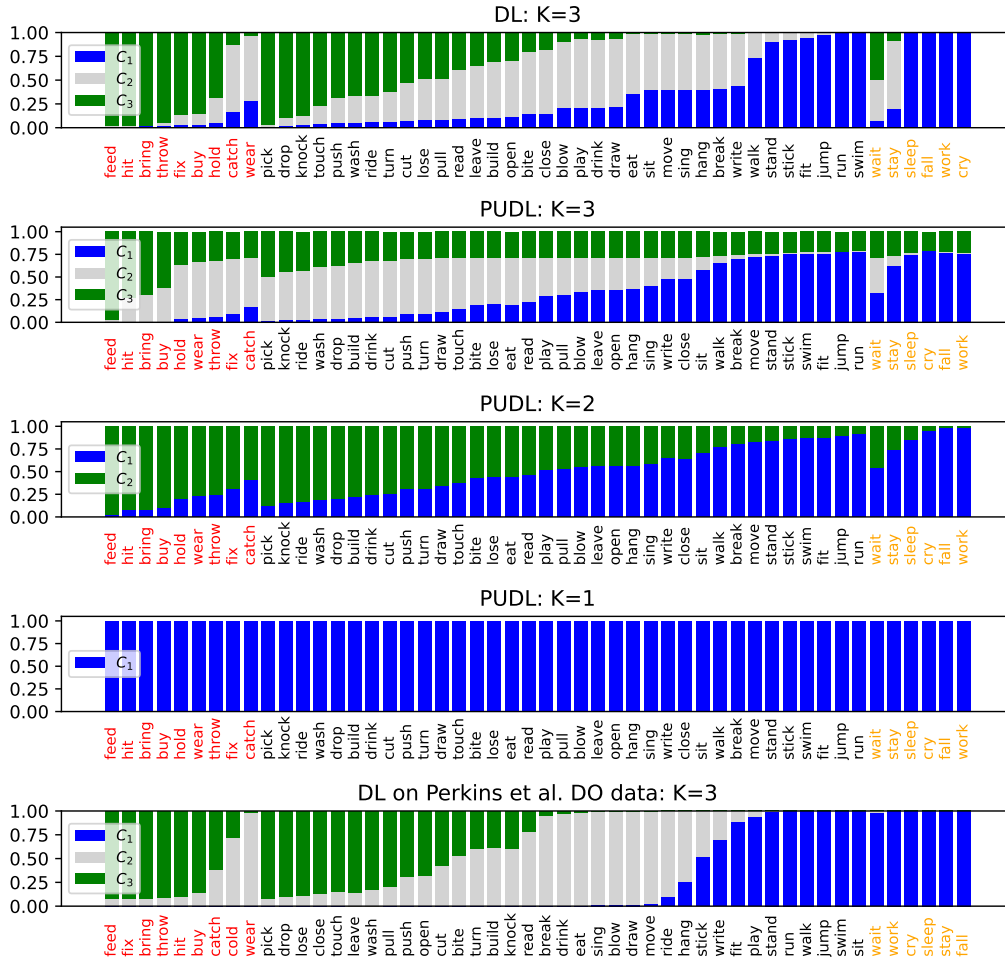


Figure 3: Posterior distributions over verb categories T .

$K = 1$ is shown in the second lowermost panel of Figure 2; its true observable PDO rates are concentrated around large values. Naturally, the posterior membership for each verb is 1 in the only available category C_1 , as shown in the second lowermost panel of Figure 3. For completeness, the estimated EB prior and posterior memberships for $K = 2$ are provided in the middle panel of Figure 2 and 3, respectively.

Intuitively, the preference for $K = 1$ indicated by BIC suggests that the pragmatically-informed learner infers that the observations are coming from a single common source, rather than two or three clusters. Capable of pragmatic reasoning about the question-answer relation, the learner made an impressive progress by recognizing verbs like ‘throw’ in (1) are more transitive than it previously thought they would be. However, the learner at the assumed developmental stage is still potentially misguided by the data like (2b) for intransitive verbs, making

an incorrect inference that verbs like ‘wait’ can occur with a direct object. Consequently, the learner undergoes the overregularizing stage, where it perceives all verbs as belonging to one category, i.e., a category with high true observable DO rates.³ This explains why $K = 1$ is preferred when the model is asked to cluster 50 verbs into K -many categories. Once the learner resolves the PP vs. NP distinction at a later stage of development, possibly after 19 months old given the experimental results in Lidz et al. (2017), we expect the result for the PUDL $K = 3$ to be more clearly separated than the DL $K = 3$, showing more progress toward deterministic knowledge. We leave the experimentation with the PUDL augmented by the PP vs. NP resolution for future research.

³It is possible to interpret this single category as either transitive or alternating. The upshot is that the learner would infer that verbs are followed by a direct object with a high probability.

6 Comparison with a filtering model

In this section, we compare our distributional learner with a filtering-based distributional learner, proposed by Perkins et al. (2022).

Filtering model The filtering-based learner identifies and filters out the inherent noise in the overt DO data, such as (1b). Assuming deterministic hypotheses of 0% DO rate, 100% DO rate, and 0-100% DO rate for intransitive, transitive, and alternating categories, respectively, the model incorporates filtering as inductive bias, allowing it to arrive at accurate generalizations only by looking at the rates of overt objects following verbs. What sets this approach apart from other proposals on filtering is that the learner operates without predetermined understanding of which data is misleading in terms of verb transitivity. All it assumes is a certain amount of noise in the data, acknowledging the presence of erroneous parses. The key insight of Perkins et al. (2022) is that the learner confronts the complex transitivity learning problem by filtering out these erroneous parses without necessarily knowing that the data such as (1b) and (2b) are non-basic clauses.

The filtering learner assumes that there are three transitivity categories $\{C_t, C_a, C_i\}$ (transitive, alternating, and intransitive) with equal prior weights. The transitivity T_i of each verb $i \in \{1, 2, \dots, V\}$ is distributed as:

$$T_i \sim \text{Uniform}(\{C_t, C_a, C_i\}).$$

Depending on the transitivity category, the verb’s true DO rate θ_i is drawn from known deterministic values or a known distribution:

$$\theta_i | T_i \sim \begin{cases} \delta_{(1)}, & \text{if } T_i = C_t \\ \text{Uniform}([0, 1]), & \text{if } T_i = C_a \\ \delta_{(0)}, & \text{if } T_i = C_i \end{cases}$$

This modeling choice encodes the deterministic hypothesis space in which there is a known category that always has a DO (‘transitive’) and another known category that never has a DO (‘intransitive’). By contrast, the categories in our model have DO rates from a flexible Beta distribution, not tied to specific values.

Notice also the difference in the definition of the parameter θ_i as a verb’s *true DO rate* in their modeling versus a verb’s *true observable DO rate* in ours. The reason behind our learner’s modeling

true observable DO rate, not true DO rate, is because our learner does not have prior knowledge about the deterministic hypotheses. Our learner is purely distributional; all the input they receive, including the utterances that we described as “misleading” above, are potentially signals that drive transitivity learning. In this regard, it is *true observable DO rate*, not *true DO rate*.

On the other hand, the filtering learner explicitly models the “misleading” part of data as noise. First, there is a parameter ϵ for the probability of an erroneous parse, which is distributed as a uniform distribution:

$$\epsilon \sim \text{Uniform}([0, 1]).$$

Second, there is another parameter δ for the probability of generating a DO in error, which is distributed as a uniform distribution:

$$\delta \sim \text{Uniform}([0, 1]).$$

Third, there is a sentence-level “input filter” $e_{i,j}$; $e_{i,j} = 1$ means the observation $X_{i,j}$ is generated from erroneous parsing. The input filter is modeled as a Bernoulli distribution:

$$e_{i,j} | \epsilon \sim \text{Bernoulli}(\epsilon).$$

Lastly, the overt DO observation $X_{i,j}$ is modeled as a mixture of the two Bernoulli distributions with success probability θ_i and δ . $X_{i,j} = 1$ means the sentence j of verb i has a DO.

$$X_{i,j} | \delta, \theta_i, e_{i,j} \sim \begin{cases} \text{Bernoulli}(\theta_i), & \text{if } e_{i,j} = 0 \\ \text{Bernoulli}(\delta), & \text{if } e_{i,j} = 1. \end{cases}$$

The filtering-based model is illustrated in the right panel of Figure 1.

Data For comparison, we present our DL’s performance on the DO data reported in Perkins et al. (2022). Note that although we follow their list of fifty verbs and use the same corpora in our analysis, the exact total count and sample DO rates are different. Specifically, the DO rates tend to be higher in our dataset because we assume that our learner hasn’t yet resolved the NP vs. PP distinction. By contrast, Perkins et al. (2022) define the overt DO as “right NP sisters of V”, which suggests that their learner can distinguish PPs from NP objects.

Result Our DL’s estimated hyperparameters (α_k, β_k) for EB Beta priors are (0.91, 8.91), (5.66, 3.70), and (30.39, 6.14) for the categories C_1 , C_2 , and C_3 . The means are 0.09, 0.60, and 0.83; the densities are shown in the lowermost panel of Figure 2, and each verb’s posterior memberships in the lowermost panel of Figure 3.

We find that our posterior membership results closely align with Figure 2 of Perkins et al. (2022). The successful verb transitivity learning reported in Perkins et al. (2022) has been attributed to the filtering mechanism, a type of inductive bias that enforces a deterministic hypothesis space. Our learner, in contrast, does not entertain a restricted hypothesis space to start with, which suggests that pure distributional learning is enough to replicate successful transitivity learning. We also highlight that our learning algorithm is simpler and more efficient than the filtering algorithm, with the runtime being less than a second.

7 Conclusion

We introduced Pragmatics-Utilizing Distributional Learner (PUDL) to model verb transitivity learning, assuming the grammatical knowledge typical of 15-month-old English learners. PUDL integrates learners’ pragmatic reasoning, particularly the realization that utterances such as ‘What did Alex throw?’ are information-seeking questions, leading in turn to the inference that this type of object wh-questions would lack a direct object, i.e., the *answer* to the question being asked. These neutral object wh-questions do not confuse pragmatically informed learners of verb transitivity, even though ‘throw’, in principle, is a transitive verb that requires a direct object. The nuanced pragmatic reasoning prompts learners to adjust their initial generalization closer to adult grammar in the domain of transitive verbs. However, the proposed pragmatic knowledge alone is insufficient to handle the noisy data in the domain of intransitive verbs. Specifically, we predicted a developmental trajectory characterized by a temporary overregularization stage, where learners generalize all verbs into a single category in terms of transitivity due to difficulty in distinguishing PP adjuncts from NP arguments. Once the PP and NP distinction is resolved⁴, possibly

⁴For instance, see Bergen et al. (2022) for recent computational modeling work on how learners differentiate between arguments and adjuncts based on distributional information. The current paper does not depend on exactly which model is adopted for the NP argument-PP adjunct resolution, as be-

after 19 months of age, as suggested by Lidz et al.’s (2017) behavioral studies, we anticipate the resolution of overgeneralization and significant progress in the intransitive domain as well, which we leave for future research. It remains to be demonstrated by behavioral experiments whether children at this critical period indeed exhibit overregularization, categorizing both the transitive verb ‘throw’ and the intransitive verb ‘wait’ into the same category in terms of transitivity. Nevertheless, we have shown that the proposed purely distributional models, Distributional Learner (DL) and Pragmatics-Utilizing Distributional Learner (PUDL), which operate confidently with received data, are as promising as an alternative distributional model that considers deterministic hypothesis space and filters out a portion of input as noise.

Acknowledgements

We thank Laurel Perkins for her invaluable guidance and discussions, which were instrumental in shaping this work, and the three anonymous reviewers for their constructive and helpful feedback. We are also grateful to the participants of the Fall 2023 UCLA graduate course on grammatical development for inspiring discussions on learning.

References

- Alison C Austin, Kathryn D Schuler, Sarah Furlong, and Elissa L Newport. 2022. Learning a language from inconsistent input: Regularization in child and adult learners. *Language Learning and Development*, 18(3):249–277.
 - Leon Bergen, Edward Gibson, and Timothy J O’Donnell. 2022. Simplicity and learning to distinguish arguments from modifiers. *Journal of Language Modelling*, 10.
 - Roger Brown. 1973. *A first language: The early stages*. Harvard University Press.
 - Daniel Buring and Christine Gunlogson. 2000. Aren’t positive and negative polar questions the same?
 - Donka F Farkas and Floris Roelofsen. 2017. Division of labor in the interpretation of declaratives and interrogatives. *Journal of Semantics*, 34(2):237–289.
 - Annie Gagliardi, Tara M Mease, and Jeffrey Lidz. 2016. Discontinuous development in the acquisition of filler-gap dependencies: Evidence from 15- and 20-month-olds. *Language Acquisition*, 23(3):234–260.
- havioral studies suggest that this differentiation is observed at least after 19 months of age, which is later than the developmental stage assumed throughout this paper.

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.

Jeffrey Lidz, Aaron Steven White, and Rebecca Baier. 2017. The role of incremental parsing in syntactically conditioned word learning. *Cognitive Psychology*, 97:62–78.

Brian MacWhinney. 2000. The childe project: Tools for analyzing talk: Volume i: Transcription format and programs, volume ii: The database.

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 Jeffrey Lidz. 2021. Eighteen-month-old infants represent nonlocal syntactic dependencies. *Proceedings of the National Academy of Sciences*, 118(41):e2026469118.

Melanie Soderstrom, Megan Blossom, Rina Foygel, and James L Morgan. 2008. Acoustical cues and grammatical units in speech to two preverbal infants. *Journal of Child Language*, 35(4):869–902.

Patrick Suppes. 1974. The semantics of children's language. *American psychologist*, 29(2):103.

Virginia Valian. 1991. Syntactic subjects in the early speech of american and italian children. *Cognition*, 40(1-2):21–81.

Meaning-Informed Low-Resource Segmentation of Agglutinative Morphology

Caleb Belth

University of Utah
caleb.belth@utah.edu

Abstract

Morphological segmentation is both an interesting acquisition problem and an important task for natural language processing. Most current computational approaches either use supervised machine learning—which tends to lead to the best-performing models—or operate over bare surface forms of words. However, the empirical conditions of language acquisition seem to fall somewhere in between: children do not have access to pre-segmented input, yet their knowledge of morphological structure develops alongside semantic knowledge. Inspired by this, we suggest a simple model for low-resource segmentation of agglutinative morphology. The model is based on the idea that agglutination tends to mark one meaning per form. It is unsupervised, but is able to exploit features to identify how differences between closely-related surface forms are marked. Trained on hundreds to a few thousand words from languages with agglutinative morphology, the resulting model outperforms an unsupervised model that does not exploit such features, and in some settings even outperforms a supervised model trained on both features and ground-truth segmentations.

1 Introduction

One of the challenges of language learning is to identify the meaning-bearing units—that is *morphological segmentation*. Segmentation has also been important to natural language processing for decades (Kurimo et al., 2010; Batsuren et al., 2022), and continues to be due to the usefulness of subword units for prominent tasks like neural language modeling and machine translation (Sennrich et al., 2016; Kudo, 2018; Brown et al., 2020; Pan et al., 2020).

The problem presents a particular challenge in agglutinative languages, where several grammatical features may be expressed by stringing together affixes. For example, the Hungarian noun

ház ‘house’ is combined with a possessive suffix *-aink* and essive case suffix ‘ban’ to form the word *házainkban* ‘in our houses’ (example from Ladányi et al. 2020, p. 191). Moreover, agglutination occurs in many low-resources languages (Moen et al., 2021; Downey et al., 2022), and occurs alongside phonological processes like vowel harmony, which lead to alternation in the form that a given affix is realized as (Ladányi et al., 2020). For example, the Hungarian essive suffix is realized as *-ben/-ban* depending on the backness of the vowel to its left, as in *szekrényben* ‘in the cupboard’ and *barlangban* ‘in the cave’ (examples from Ladányi et al. 2020, p. 192).

Some approaches to segmentation are supervised, meaning that they learn from segmented training data. For example, the winner of the 2022 SIGMORPHON (Batsuren et al., 2022) segmentation challenge was a sequence-to-sequence transformer model (Peters and Martins, 2022). Other approaches—often preferred due to not requiring annotated training data—are unsupervised approaches, which are usually trained on bare surface forms (e.g., Uchiumi et al. 2015; Xu et al. 2020).

Our approach takes inspiration from language acquisition, where children show evidence of an ability to analyze words in morphologically-complex ways, segmenting them into distinct subunits or morphemes (Marquis and Shi, 2012; Mintz, 2013; Ladányi et al., 2020; Kim and Sundara, 2021). For example, Ladányi et al. (2020) demonstrated that when the common Hungarian suffix *-ban* was attached to a nonce stem (e.g., *pür-ban*), 15mo Hungarian-learning children indeed analyzed such nonces as suffixed words, as evidenced by their ability to later recognize the stem in bare form (see § 2.1 for more discussion).

We suggest that one mechanism useful to morphological segmentation in agglutinative languages could be the ability to recognize pairs of closely-related word forms, and then infer sim-

ple differences between each pair. For example *tanárok**nak*, the Hungarian plural (PL) dative (DAT) of ‘teacher’, differs from *tanárok* ‘teachers’ in only one feature (case), and the former can be derived from the latter by suffixing *-nak*. This provides the learner evidence that DAT can be marked by the suffix *-nak*. Moreover, if the learner knows from other pairs like *tanár/tanárok* ‘teacher’/‘teachers’ that plurals are also marked by suffixation, then they can infer evidence that the DAT suffix is ordered after the plural suffix.

In this paper, we implement this proposal as a simple segmentation model, which uses morphological features to identify closely-related word pairs, from which it infers the concatenative operations that the language uses to mark those features. This approach offers the possibility of improvement over unsupervised approaches that operate over only surface forms, while simplifying the data-annotation demands of supervised approaches needing ground-truth segmentations. For example, Unimorph 3.0 (McCarthy et al., 2020) contains morphological features for 169 languages, but segmentations—via MorphyNet (Batsuren et al., 2021)—for only 15.

When trained on 500-10,000 words, the model achieves 72-100% accuracy segmenting test words in Finnish, Hungarian, Mongolian, and Turkish, out-performing the unsupervised model Morfessor 2.0 (Virpioja et al., 2013) and, in a majority of cases, a supervised neural comparison model. These results suggest that the model could be useful for segmenting low-resource agglutinative languages, since producing a small number of morphological-feature-annotated word forms is often easier than producing ground-truth segmentations, and such feature annotations yield large improvements over segmentations based on bare surface forms.

2 Model

2.1 Cognitive Motivation

Our model is motivated by experimental findings from child language acquisition. We are primarily concerned with the empirical promise of the model to segment agglutinative morphology, but in § 5 we discuss the extent to which we think the model is itself revealing about the mechanisms of the acquisition of morphological structure.

Marquis and Shi (2012) found that 11mo-old French-learning infants could perceive nonce

words suffixed with the frequent French verbal suffix *-e* as related to their bare stems. This ability was not attested when an unfamiliar suffix *-u* was attached to nonce stems, suggesting that the infants were decomposing the nonces into stem and affix units rather than recognizing phonological overlap. At 15mo, Mintz (2013) found similar results for the English suffix *-ing*, and likewise Ladányi et al. (2020) for the Hungarian essive suffix *-ban/-ben*. The ability to relate forms was unperturbed by the vowel-harmony-induced alternation between suffix forms. Thus, given Hungarian’s agglutinative morphology, the ability to relate inflected forms to their stems seems to develop even in the presence of agglutination and alternation.

Many of these results also suggest that the ability to relate closely-related forms may begin developing prior to children acquiring the function of morphemes. For example, Marquis and Shi (2012) found that presenting infants with many nonce words inflected with an unfamiliar suffix, they would begin to relate the inflected nonces to their stems. Moreover, Kim and Sundara (2021) found that the ability emerges for at least some English suffixes (*-s*) as early as 6mo, even when the nonces are presented without referential context, which they take as evidence that the ability begins developing without dependence on meaning.

Together, these studies suggest that infants can relate (concatenatively) inflected forms to their stems, and that this ability at least begins to emerge prior to children learning the function of morphemes. Payne (2022, 2023) has proposed that this early segmentation ability could allow children to identify *collisions*, which are instances of stems appearing in multiple inflected forms. Payne argues that these collisions provide evidence to the learner about what morphological features are marked in the language being acquired, via Clark (2014)’s observation that differences in form are indicative of differences in meaning. Payne’s proposal, implemented as an explicit learning model, accurately matches developmental findings.

Given the well-attested ability of infants to relate word forms that differ in a single affix and the plausibility of Payne (2023)’s hypothesis about how learners can use this to discover what morphological features are marked, learners could then use the differences between related word forms and their developing knowledge of marked morphological features to identify what subparts of words correspond to these marked features—that is to pro-

duce meaning-informed segmentations. This idea forms the basis of our proposed model, which we present next.

2.2 Input

The model’s input training data is a collection of $\langle w, r, f \rangle$ words (triples), where w is the word’s surface form, r is the word’s root meaning, and f is a set of morphological features marked in the word. Notably, r is only the root *meaning* and not the root form. An example input is (1), which we will use as a running example.

- (1) a. (*tanár*, TEACHER, { })
- b. (*tanárok*, TEACHER, { PL })
- c. (*tanároknak*, TEACHER, { PL, DAT })
- d. (*személy*, PERSON, { })
- e. (*személynek*, PERSON, { DAT })

2.3 Learning Algorithm

The model, which we call MIASEG [ˈmi.ə.sɛɡ] for *Meaning-Informed Agglutinative Segmentation*, learns from the input described above by identifying closely-related words and inferring the concatenative difference between their forms as a candidate marking of the feature difference between the words. MIASEG considers two words to be *closely-related* if they have the same root meaning and one has all the features of the other plus one.

Thus, a paradigm P_m corresponding to a root meaning m is represented as the set of input triples whose root meaning equals m , (2).

$$(2) \quad P_m \triangleq \{ \langle w, r, f \rangle : r = m \}$$

For example, the paradigm P_{TEACHER} contains (1a)-(1c). This is shown in (5; step 1). MIASEG then computes, for each paradigm, the closely-related words in the paradigm—namely those where one word has all the features of the other plus one (3).

$$(3) \quad c(P_m) \triangleq \{ \langle w_i, r_i, f_i \rangle, \langle w_j, r_j, f_j \rangle \in P_m : |f_i \cup f_j \setminus f_i \cap f_j| = 1 \}$$

Thus $c(P_{\text{TEACHER}})$ returns the pairs (1a)-(1b) and (1b)-(1c). For each of these pairs (5; step 2), MIASEG computes the string difference between the word forms w_i and w_j (5; step 3) and posits the difference as one way of marking the feature that differs between the two words (5; step 4). MIASEG represents this inference as a triple of the form (4), where $\phi = |f_i \cup f_j \setminus f_i \cap f_j|$ is the marked feature, Δ is the concatenative difference between w_i and

w_j , and t specifies whether the difference is a suffix (i.e., comes at the right edge) or a prefix (i.e., comes at the left edge).

$$(4) \quad \langle \phi, \Delta, t \rangle$$

For example, the difference between (1b)-(1c) is the presence of an ending *-nak* in (1c), which has the additional feature DAT. Thus, MIASEG infers that the suffix *-nak* is one way of marking DAT: $\langle \text{DAT}, \text{nak}, \text{SUFF} \rangle$. MIASEG also stores the number of times the difference has been inferred as a marking of the feature (i.e., the frequency of each triple), for prioritizing among multiple analyses during segmentation (§ 2.4). Moreover, because both (1b) and (1c) have the feature PL, MIASEG tabulates that the PL marker probably comes before the DAT marker.

At a different iteration of the loops, MIASEG will find the difference between (1d) and (1e) to be *-nek* and MIASEG will learn that this is another way to mark DAT. Thus, the markings inferred by MIASEG are effectively allomorphs of the morphemes corresponding to each marked feature. The resulting segmentations could be used as the input to a method like Belth (2023a)’s, which constructs underlying forms for morphemes based on surface alternation.

Once the for loops are complete, MIASEG infers a global ordering of features (5; step 6) by creating a directed graph, where each feature forms a node and an edge is formed from f_i to f_j whenever it was inferred that f_i must come before f_j (e.g., PL \rightarrow DAT). The graph is then topologically sorted, which yields a total linear ordering of the features such that any orderings encoded in the graph edges are preserved in the linear ordering (Cormen et al., 2009, p. 612).¹

- (5) **Input:** Set of $\langle w, r, f \rangle$ triples
 1. **For** each paradigm P_m in data **do**
 2. **For** pair in $c(P_m)$ **do**
 3. — Find Δ between w_i and w_j
 4. — Posit Δ as marking of $f_i \cup f_j \setminus f_i \cap f_j$
 5. — Tabulate implied feature orderings
 6. Infer global ordering of features

We discuss the strengths and limitations of this algorithm in § 5. The code is available at <https://github.com/cbelth/miaseg>.

¹Extensions may be necessary for languages with variable morpheme order, as this would introduce cycles into the graph. In the current implementation, if the ordering $f_i \rightarrow f_j$ and $f_j \rightarrow f_i$ are both inferred, only the ordering that was inferred the most times at line 5 is inserted into the graph.

2.4 Segmentation

Once the ways in which morphological features can be marked, and the ordering among them, are inferred and recorded, the model can segment words—either the words from which it made these inferences or new (test) words.

Segmentation takes as input a surface form, w (e.g., *csapatoknak*), and set of features f (e.g., {PL, DAT}). MIASEG iterates (6; step 1) over each feature in f in an order consistent with the ordering inferred during training (5; step 6)—left-to-right for prefixes and right-to-left for suffixes (e.g., DAT then PL since PL \rightarrow DAT was inferred during training).

For each feature, the model looks up the ways in which it was marked in the training data (6; step 2), and tries each marking until one matches the end (for suffixes) or beginning (for prefixes) of w . The markings are considered in descending order of length, using the number of times the marking was attested in the training data as a tie breaker for equal-length matches. When a match is found, the matching ending is separated from the word as a morpheme. For example, DAT was marked as *-nak* and *-nek* in the training data, and *csapatoknak* ends in *-nak*, so *nak* is separated from the word to form *csapatok-nak*. The segmentation algorithm then proceeds to the next feature. For example, the model then looks at the ways in which PL can be marked for a match at the ending of *csapatok*, which will find *-ok*, resulting in *csapat-ok-nak*.

If at any point no attested marking of a feature matches (6; step 5), to prevent this from blocking further segmentation, MIASEG separates k segments from w , where k is the most common length of attested markings (for example $k = 1$ for a feature with attested markings $\{a, e, ja\}$).

- (6) **Input:** $\langle w, f \rangle$ pair
 1. **For** feat in f (ordered) **do**
 2. — **For** attested marking of f **do**
 3. — **If** marking matches edge of w **then**
 4. — Separate marking from w
 5. — **If** no attested marking matched **do**
 6. — Separate k segments from w
 7. **Return** segmented w

3 Evaluation

Our evaluation attempts to test the effectiveness of the model at segmenting agglutinative languages in relatively low-resources settings, where only hun-

Table 1: Dataset Sizes

Fin	541,198
Hun	613,549
Mon	11,215
Tur	18,333

dreds to a few thousands words are available for training.

3.1 Data

We collected data for Finnish (Fin), Hungarian (Hun), Mongolian (Mon), and Turkish (Tur), all languages with a substantial amount of agglutinative morphology. The languages come from three language families: Finnish and Hungarian are Uralic languages, Mongolian is a Mongolic language, and Turkish is a Turkic language. For all datasets except Turkish, we followed Batsuren et al. (2022) in using data from MorphyNet (Batsuren et al., 2021), which has canonical segmentations extracted from Wiktionary. For Turkish, we followed Belth (2023a,b, 2024) in using the corpus created for MorphoChallenge (Kurimo et al., 2010). We used Çöltekin (2010, 2014)’s publicly-available finite state morphological analyzer to generate morphological analyses.² The analyzer is designed for Turkish, and is similar to the approach used by MorphoChallenge to generate ground-truth analyses. For simplicity, we decided to look only at nouns for this paper. For each dataset, we extracted all nouns where we could unambiguously convert the canonical segmentation to a surface segmentation (Cotterell et al., 2016). The resulting dataset sizes are shown in Tab. 1.

We also collected corpus frequency information for each word in each dataset. For Finnish and Mongolian, we used the very large monolingual datasets aggregated by Conneau et al. (2020); Wenzek et al. (2020) from the 2018 CommonCrawl, counting the frequency of each word in the corpus. For Hungarian, we used the Hungarian Web Corpus (Halácsy et al., 2004) frequency file. Any word in our datasets that did not occur in these web corpora we assumed to be low frequency (given the extremely large size of the web corpora); we assigned them frequency of 1. The Turkish dataset already contained frequency information.

²<https://github.com/coltekin/TRmorph>

3.2 Setup

We discuss comparison models in § 3.2.1 and the training and evaluation procedures in § 3.2.2.

3.2.1 Comparison Models

We compare `MIASEG`, which is unsupervised but requires data be annotated with morphological features, to `MORFESSOR`, which is an unsupervised model that segments bare surface forms, and to `TRANSFORMER`, a supervised transformer-based encoder-decoder sequence to sequence (seq2seq) model that learns from segmented training data that is annotated with the same morphological features that `MIASEG` uses.

For `MORFESSOR`, we used the `Morfessor 2.0` model (Virpioja et al., 2013), which is available as a Python package.

`TRANSFORMER` is the name of a supervised neural seq2seq model that we apply to the task. Neural seq2seq models have had success at many morphological problems, including the 2022 `SIGMORPHON` (Batsuren et al., 2022) challenge on morphological segmentation, where Peters and Martins (2022)’s `DeepSPIN-3` model achieved the best-overall performance on the word-level task. However, to our knowledge, the code for `DeepSPIN-3` is not publicly available, and the model does not incorporate morphological features. On the other hand, neural seq2seq models consistently perform well at the recurring `SIGMORPHON` morphological inflection task (see Kodner et al. 2022 for a recent iteration of the task), and these models commonly incorporate morphological features directly into the model, due to their importance to the inflection task (e.g., Makarov and Clematide 2018; Wu et al. 2021).

Thus, we follow Wu et al. (2021) in using a transformer-based encoder-decoder architecture, which includes both morphological features and word characters in the model’s vocabulary. We describe the model’s architecture in more detail below (§ 3.2.2).

3.2.2 Training and Evaluation

While unsupervised models like `MORFESSOR` and `MIASEG` can be evaluated on how well they segment the training data since they receive no information about the ground-truth segmentations during training, we wish to compare performance to the supervised setting (represented by `TRANSFORMER`), which necessitates evaluating on a held-out test set. Consequently, we chose to evaluate all

three models on held-out test sets.

In relatively low-resource settings, as well as in child language acquisition, higher-frequency words are more likely to be represented than lower-frequency words. To approximate such a situation, we chose to sample training words weighted by frequency. We evaluated at three different training sizes: 500, 1000, and 10000. For each training size, we ran each model on 10 samples with different random seeds. Every word not included in the training sample was included in the held-out test set.

On each of the 10 random seeds, we tuned `TRANSFORMER`’s hyperparameters using a grid search sweep. To do so, we made a random 80%/20% split of the training data, and trained the model with each hyperparameter combination on the 80% part of the split; we evaluated accuracy on the remaining 20%. We chose the hyperparameter combination that yielded the best accuracy on the 20%, remerged the 80%/20% split into the full training set, and then trained a new model with that hyperparameter combination on the entire training split. The hyperparameters we considered were those in (7), yielding 48 combinations.

- (7) Embedding Dimension $\in \{256, 512\}$
Dropout = $\in \{0.1, 0.3\}$
Batch Size = $\in \{32, 128, 256\}$
Number of Enc. & Dec. Layers = $\in \{1, 2\}$
Number of Attention Heads = $\in \{4, 8\}$

We evaluate all models in terms of precision, recall, F1, and accuracy. Precision measures, out of all predicted morphemes (across the entire test set), what fraction are actually morphemes. Recall measures, out of all morphemes, what fraction are predicted. F1 measures the harmonic mean of precision and recall. Accuracy measures the fraction of test items that are correctly segmented.

3.3 Results

The results (F1 and accuracy)³ are shown in Tab. 2. `MIASEG` outperforms the unsupervised `MORFESSOR` by a large margin on all datasets, and even outperforms the supervised `TRANSFORMER` model on 3/4 datasets. Importantly, the accuracy—not just the F1—is fairly high in absolute terms, even at a training size of only 1000 words. This means that a large majority of the test words are correctly segmented.

³We report the precision and recall values going into the F1 scores in Tab. 4 in the appendix.

MIASEG’s performance is noticeably worse for Finnish than the other datasets, though it still performs competitively with the supervised TRANSFORMER and still outperforms the unsupervised MORFESSOR baseline. The primary reason for this is that the NOM plural is usually marked with [-t], but in the other noun cases (except ACC), it is marked with [-i]. For example *auto-t* is the plural NOM of ‘car’, while *auto-i-ssa* is the plural IN+ESS. Because case markers come after plural markers, [-i] never occurs at a word boundary, so MIASEG never recognizes it as a possible plural marker. This accounts for over 80% of MIASEG’s errors on Finnish.

The reason MIASEG is able to achieve such high accuracy on Mongolian is that the Unimorph data from which it was derived only contains nouns with a single affix, which marks 1 of 7 cases (GEN, ACC, DAT, ABL, INS, COM, VOC). Thus, once the model has been exposed to sufficient nouns to have seen all allomorphs of those case suffixes, it is able to achieve perfect segmentation of the limited set of nouns. In contrast, all other evaluation languages have chains of multiple affixes in their respective datasets. We note though, that the simplicity of the task for Mongolian is also true for MORFESSOR and TRANSFORMER, which never achieve the same performance on Mongolian.

A few randomly-selected example segmentations are shown in Tab. 3 (we excluded Mongolian since the data only contained single affixes). The first example from Turkish, where MIASEG segmented *gazetelerinizi* ‘newspapers-PL-PSS2P-ACC’ as *gazete-ler-iniz-i* demonstrates that MIASEG is able to segment multiple affixes, having inferred that plurality is marked first and case last.

3.3.1 Error Analysis

We performed error analysis of MIASEG for each language at the training size of 10K. In Finnish, > 99% of the errors are due to failing to find a match for a suffix, probably due to some suffixes not occurring at a word boundary. As discussed above, this aspect of the non-NOM PL allomorphs led to 80% of MIASEG’s errors on Finnish.

For Hungarian, 58% of errors are of the same type as Finnish. 26% of the errors involve shifting a morpheme boundary to the left (e.g., *tolvaj-a* vs. **tolva-ja*) and 16% are due to shifting a morpheme boundary to the right (e.g., *ezán-jaik* vs. **ezánj-aik*). For Turkish, >0.99% of errors involve shifting a morpheme boundary to the left.

The relative prevalence of errors involving shifting a morpheme boundary to the left is likely because MIASEG considers the forms that a feature has been marked with (6; step 2) in descending order of length. Thus, if two forms match (e.g., both *aj* and *a* are allomorphs of the PSS3S;SG suffix and match the end of *tolvaja* ‘thief-PSS3S;SG’), the longer will be chosen. If the shorter was the correct form, the morpheme boundary is effectively shifted left.

These error patterns suggest that promising areas for improvement would be handling affixes not appearing at word boundaries and improving the heuristic preference for the longest matching marking during segmentation (6; step 2). Note that this analysis considered errors at the word level, meaning that we identified one of potentially multiple reasons for each incorrectly-segmented word. Thus, of the errors in Finnish (>0.99%) and Hungarian (58%) attributed to failing to find a match for a suffix, it is possible that some also had morpheme boundaries shifted left or right.

4 Prior Work

Unsupervised segmentation methods include Minimum Description Length (MDL) models (e.g., Goldsmith 2001). A prevelant model, at least as a baseline, is Morfessor (Creutz and Lagus, 2002) and derivations of it (Creutz and Lagus, 2005, 2007; Virpioja et al., 2013). Bayesian models are often successful, though many were developed in the context of word segmentation (e.g., Goldwater et al. 2009). Neural models have also been employed, usually using a self-supervised task like segmental language modeling for training (Sun and Deng, 2018; Downey et al., 2022; Wang and Zheng, 2022).

Like MIASEG, some prior unsupervised approaches explicitly model morphological paradigms (Goldsmith, 2001; Xu et al., 2018, 2020). Moreover, we are not the first approach to consider meaning, along with form, for segmentation. Prior approaches learn word embeddings to represent semantic information through distributional information (Schone and Jurafsky, 2001; Soricut and Och, 2015; Narasimhan et al., 2015). In contrast, we use morphological features from Unimorph (McCarthy et al., 2020), not word embeddings, which can be data-intensive to train.

Some models attempt to achieve broad typological coverage. For instance, Morfessor (Creutz and

Table 2: F1 (harmonic mean of precision and recall) and accuracy of models. MIASEG, which is our model, outperforms MORFESSOR, which is unsupervised and cannot make use of morphological features, on all datasets and data sizes. Moreover, on 3/4 datasets, MIASEG outperforms TRANSFORMER, which trains in a supervised fashion on both ground-truth segmentations and morphological features.

		500		1000		10000	
		F1	Acc	F1	Acc	F1	Acc
Fin	MIASEG	0.57 ± 0.03	0.48 ± 0.03	0.69 ± 0.04	0.61 ± 0.04	0.79 ± 0.00	0.72 ± 0.00
	MORFESSOR	0.27 ± 0.03	0.18 ± 0.02	0.27 ± 0.02	0.17 ± 0.01	0.19 ± 0.01	0.05 ± 0.00
	TRANSFORMER	0.63 ± 0.04	0.48 ± 0.05	0.73 ± 0.03	0.61 ± 0.04	0.90 ± 0.03	0.83 ± 0.05
Hun	MIASEG	0.41 ± 0.05	0.32 ± 0.05	0.63 ± 0.07	0.56 ± 0.07	0.94 ± 0.01	0.93 ± 0.02
	MORFESSOR	0.19 ± 0.05	0.12 ± 0.03	0.32 ± 0.04	0.18 ± 0.03	0.32 ± 0.01	0.13 ± 0.01
	TRANSFORMER	0.48 ± 0.03	0.27 ± 0.04	0.61 ± 0.02	0.43 ± 0.03	0.82 ± 0.06	0.72 ± 0.09
Mon	MIASEG	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	MORFESSOR	0.55 ± 0.03	0.48 ± 0.04	0.49 ± 0.03	0.39 ± 0.03	0.91 ± 0.01	0.89 ± 0.02
	TRANSFORMER	0.79 ± 0.04	0.73 ± 0.05	0.93 ± 0.02	0.90 ± 0.03	0.98 ± 0.01	0.97 ± 0.01
Tur	MIASEG	0.83 ± 0.00	0.81 ± 0.00	0.94 ± 0.01	0.92 ± 0.01	0.96 ± 0.00	0.94 ± 0.00
	MORFESSOR	0.47 ± 0.04	0.32 ± 0.04	0.46 ± 0.03	0.30 ± 0.03	0.54 ± 0.01	0.36 ± 0.01
	TRANSFORMER	0.75 ± 0.03	0.60 ± 0.04	0.86 ± 0.03	0.77 ± 0.05	0.94 ± 0.01	0.90 ± 0.03

Table 3: A few randomly-selected segmentations from MIASEG.

	Word & Features	Predicted	Expected	
Tur	<i>gazetelerinizi</i> (PL;PSS2P;ACC)	<i>gazete-ler-iniz-i</i>	<i>gazete-ler-iniz-i</i>	✓
Fin	<i>ilmaperspektiivein</i> (INS;PL)	<i>ilmaperspektiive-in</i>	<i>ilmaperspektiive-in</i>	✓
Hun	<i>hátraküldésünk</i> (PSS1P;SG)	<i>hátraküldés-ünk</i>	<i>hátraküldés-ünk</i>	✓
Fin	<i>eristysselleillä</i> (PL;AT+ESS)	<i>eristyssellei-llä</i>	<i>eristysselle-i-llä</i>	✗
Tur	<i>mikroorganizmalardan</i> (PL;ABL)	<i>mikroorganizma-lar-dan</i>	<i>mikroorganizma-lar-dan</i>	✓

Lagus, 2002, 2005, 2007; Virpioja et al., 2013) can easily be applied to data from any language. Xu et al. (2020) directly leverage typology by incorporating a diverse range of morphological processes beyond affixation. The resulting model leads to strong results across typologically and phylogenetically diverse languages.

Other approaches have focused on particular typologically or phylogenetically related groups of languages. Pan et al. (2020) proposed an approach to segmenting agglutinative languages for the task of machine translation. Moeng et al. (2021) developed supervised and unsupervised approaches for morphological segmentation of Nguni Languages. Downey et al. (2022) demonstrated that training a neural model in a self-supervised task on ten Indigenous languages of the Americas that are typologically related but phylogenetically unrelated can transfer to a target language, K’iche’.

Our work is in line with the latter group, as we

focus on agglutinative morphology. We believe there are merits to both approaches. While typological coverage is an important goal, we believe focusing on mechanisms that may be useful for particular kinds of morphological structure is also of value, since languages can differ dramatically in their morphological structure. For instance, we should not necessarily expect the acquisition of agglutinative and templatic morphological processes to involve precisely the same mechanisms.

5 Conclusion and Discussion

In this work, we have proposed a model for unsupervised but morphological-feature-informed segmentation of agglutinative morphology. Our proposed model, MIASEG, takes advantage of the fact that in agglutinative morphology, a single morpheme tends to correspond to a single feature. Thus, by identifying closely-related pairs of words—i.e. words where one has exactly one fea-

ture more than the other—and inferring the concatenative difference between them, the model is able to discover the ways in which morphological features are marked. These markings are effectively the allomorphs of a given morpheme.

When trained in low resource settings of 500, 1000, or 10000 words, MIASEG achieved reasonably high accuracy and F1 scores across Finnish, Hungarian, Mongolian, and Turkish. Moreover, MIASEG outperformed the unsupervised model MORFESSOR, which operates over bare surface forms—demonstrating the value of morphological features. In a majority of settings, MIASEG also outperformed a supervised neural model that was able to exploit the same features. This suggests that MIASEG, while a simple approach, can outperform a supervised model in low-resource settings.

We find the results to be encouraging for our proposed approach to agglutinative morphology, though we acknowledge that much of the approach would require work to extend to many types of non-agglutinative morphology.

In particular, MIASEG exploits the fact that in agglutinative morphology, each morpheme tends to mark a single feature. In contrast, fusional morphological processes mark multiple features with a single morpheme, leading to its own set of learning challenges. Moreover, morphological processes also include non-concatenative stem changes, reduplication, and templatic processes.

Even among concatenative operations, Xu et al. (2020, p. 6673) point out that some languages have affixes that never appear at a word edge because the affix is always followed or preceded by another affix. Because our method depends on identifying concatenative differences between word forms that differ in a single marked feature, our model would need to be extended in order to discover such affixes. We saw this issue in Finnish, where MIASEG achieved its lowest performance due to some plural allomorphs never occurring at the edge of a word.

Our approach to segmentation takes inspiration from findings in child language acquisition (§ 2.1). We have proposed that if a learner knows which morphological features are marked in a language, the learner can use this information to identify morpheme boundaries in an approach like the one we have proposed. We intend the model for practical use in low-resource, agglutinative morphological segmentation settings and not as an acquisition model. That said, the fact that the approach is inspired by considerations of acquisition and is rea-

sonably effective makes it somewhat tantalizing to conjecture that a similar mechanism might be at play when children acquire agglutinative morphological processes. In future work, we plan to investigate this proposal more directly.

References

- Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. [The SIGMORPHON 2022 shared task on morpheme segmentation](#). In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.
- Khuyagbaatar Batsuren, Gábor Bella, and Fausto Giunchiglia. 2021. Morphynet: a large multilingual database of derivational and inflectional morphology. In *Proceedings of the 18th sigmorphon workshop on computational research in phonetics, phonology, and morphology*, pages 39–48.
- Caleb Belth. 2023a. [Towards a learning-based account of underlying forms: A case study in Turkish](#). In *Proceedings of the Society for Computation in Linguistics 2023*, pages 332–342, Amherst, MA. Association for Computational Linguistics.
- Caleb Belth. 2023b. *Towards an Algorithmic Account of Phonological Rules and Representations*. Ph.D. thesis, University of Michigan.
- Caleb Belth. 2024. [A Learning-Based Account of Phonological Tiers](#). *Linguistic Inquiry*, pages 1–63.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Eve V Clark. 2014. The principle of contrast: A constraint on language acquisition. In *Mechanisms of language acquisition*, pages 1–33. Psychology Press.
- 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.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised](#)

- cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms*. MIT press.
- Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016. A joint model of orthography and morphological segmentation. Association for Computational Linguistics.
- Mathias Creutz and Krista Lagus. 2002. **Unsupervised discovery of morphemes**. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6, MPL '02*, page 21–30, USA. Association for Computational Linguistics.
- Mathias Creutz and Krista Lagus. 2005. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Helsinki University of Technology Helsinki.
- Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):1–34.
- C. Downey, Shannon Drizin, Levon Haroutunian, and Shivin Thukral. 2022. **Multilingual unsupervised sequence segmentation transfers to extremely low-resource languages**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5331–5346, Dublin, Ireland. Association for Computational Linguistics.
- John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.
- Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2009. A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.
- Péter Halácsy, András Kornai, Németh László, Rung András, István Szakadát, and Trón Viktor. 2004. Creating open language resources for hungarian.
- Yun Jung Kim and Megha Sundara. 2021. 6-month-olds are sensitive to english morphology. *Developmental science*, 24(4):e13089.
- Jordan Kodner, Salam Khalifa, Khuyagbaatar Batsuren, Hossep Dolatian, Ryan Cotterell, Faruk Akkus, Antonios Anastasopoulos, Taras Andrushko, Aryaman Arora, Nona Atanalov, Gábor Bella, Elena Budianskaya, Yustinus Ghanggo Ate, Omer Goldman, David Guriel, Simon Guriel, Silvia Guriel-Agiashvili, Witold Kieraś, Andrew Krizhanovsky, Natalia Krizhanovsky, Igor Marchenko, Magdalena Markowska, Polina Mashkovtseva, Maria Nepomniashchaya, Daria Rodionova, Karina Scheifer, Alexandra Sorova, Anastasia Yemelina, Jeremiah Young, and Ekaterina Vylomova. 2022. **SIGMORPHON–UniMorph 2022 shared task 0: Generalization and typologically diverse morphological inflection**. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 176–203, Seattle, Washington. Association for Computational Linguistics.
- Taku Kudo. 2018. **Subword regularization: Improving neural network translation models with multiple subword candidates**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.
- 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.
- Peter Makarov and Simon Clematide. 2018. **Imitation learning for neural morphological string transduction**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882, Brussels, Belgium. Association for Computational Linguistics.
- Alexandra Marquis and Rushen Shi. 2012. **Initial morphological learning in preverbal infants**. *Cognition*, 122(1):61–66.
- Arya D McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J Mielke, Garrett Nicolai, Miikka Silfverberg, et al. 2020. Unimorph 3.0: Universal morphology. In *Proceedings of The 12th language resources and evaluation conference*, pages 3922–3931. European Language Resources Association.
- Toben H Mintz. 2013. The segmentation of sublexical morphemes in english-learning 15-month-olds. *Frontiers in psychology*, 4:24.
- Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2021. Canonical and surface morphological segmentation for nguni languages. In *Southern African Conference for Artificial Intelligence Research*, pages 125–139. Springer.
- Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

- Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. 2020. Morphological word segmentation on agglutinative languages for neural machine translation. *arXiv preprint arXiv:2001.01589*.
- Sarah Payne. 2022. *When collisions are a good thing: the acquisition of morphological marking*. Bachelor’s thesis, University of Pennsylvania.
- Sarah Payne. 2023. Contrast, sufficiency, and the acquisition of morphological marking. In *Proceedings of BUCLD*, volume 47, pages 604–617.
- Ben Peters and Andre F. T. Martins. 2022. **Beyond characters: Subword-level morpheme segmentation**. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 131–138, Seattle, Washington. Association for Computational Linguistics.
- Patrick Schone and Dan Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Second meeting of the north american chapter of the association for computational linguistics*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. **Neural machine translation of rare words with subword units**. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Radu Soricut and Franz Josef Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637.
- Zhiqing Sun and Zhi-Hong Deng. 2018. **Unsupervised neural word segmentation for Chinese via segmental language modeling**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4915–4920, Brussels, Belgium. Association for Computational Linguistics.
- Kei Uchiumi, Hiroshi Tsukahara, and Daichi Mochihashi. 2015. Inducing word and part-of-speech with pitman-yor hidden semi-markov models. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1774–1782.
- Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.
- Lihao Wang and Xiaoqing Zheng. 2022. **Unsupervised word segmentation with bi-directional neural language model**. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(1).
- Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. **CCNet: Extracting high quality monolingual datasets from web crawl data**. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.
- Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. **Applying the transformer to character-level transduction**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.
- Hongzhi Xu, Jordan Kodner, Mitch Marcus, and Charles Yang. 2020. Modeling morphological typology for unsupervised learning of language morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6672–6681.
- Hongzhi Xu, Mitch Marcus, Charles Yang, and Lyle Ungar. 2018. Unsupervised morphology learning with statistical paradigms. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 44–54.

A Example Appendix

In Tab. 4, we provide the precision and recall values for the models; these values went in to the computation of F1 scores in Tab. 2.

Table 4: Precision and Recalls for models. These correspond to the F1 scores in Tab. 2.

		500		1000		10000	
		P	R	P	R	P	R
Fin	MIASEG	0.67 ± 0.03	0.50 ± 0.03	0.77 ± 0.03	0.62 ± 0.04	0.84 ± 0.00	0.74 ± 0.00
	MORFESSOR	0.31 ± 0.03	0.24 ± 0.02	0.27 ± 0.02	0.28 ± 0.02	0.14 ± 0.01	0.28 ± 0.00
	TRANSFORMER	0.63 ± 0.04	0.63 ± 0.05	0.73 ± 0.03	0.73 ± 0.03	0.89 ± 0.04	0.90 ± 0.03
Hun	MIASEG	0.48 ± 0.05	0.35 ± 0.05	0.69 ± 0.06	0.59 ± 0.07	0.95 ± 0.01	0.94 ± 0.02
	MORFESSOR	0.24 ± 0.05	0.16 ± 0.04	0.30 ± 0.04	0.34 ± 0.04	0.26 ± 0.01	0.43 ± 0.01
	TRANSFORMER	0.49 ± 0.04	0.46 ± 0.02	0.62 ± 0.02	0.60 ± 0.03	0.83 ± 0.07	0.82 ± 0.06
Mon	MIASEG	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	MORFESSOR	0.50 ± 0.04	0.61 ± 0.02	0.41 ± 0.03	0.60 ± 0.01	0.89 ± 0.02	0.94 ± 0.01
	TRANSFORMER	0.80 ± 0.03	0.78 ± 0.05	0.93 ± 0.02	0.93 ± 0.02	0.98 ± 0.01	0.98 ± 0.01
Tur	MIASEG	0.85 ± 0.00	0.81 ± 0.00	0.94 ± 0.01	0.93 ± 0.01	0.96 ± 0.00	0.96 ± 0.00
	MORFESSOR	0.48 ± 0.04	0.47 ± 0.04	0.44 ± 0.04	0.48 ± 0.02	0.58 ± 0.01	0.50 ± 0.01
	TRANSFORMER	0.75 ± 0.03	0.75 ± 0.02	0.86 ± 0.03	0.86 ± 0.03	0.94 ± 0.01	0.94 ± 0.01

Tiers, Paths, and Syntactic Locality: The View from Learning

Kenneth Hanson

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

Abstract

Many long-distance linguistic dependencies across domains can be modeled as *tier-based strictly local* (TSL) patterns (Graf, 2022a). Such patterns are in principle efficiently learnable, but known algorithms require unrealistic conditions. In contrast, Heuser et al. (2024) present an empirically-grounded algorithm which learns syntactic islands by tracking bigrams along movement paths, but does not involve tiers. I combine the advantages of both approaches by adapting the latter algorithm to produce a TSL grammar. This method is capable of learning other syntactic blockers besides islands, and augments the typological predictions of the TSL model with a version of the Height-Locality Connection (Keine, 2019).

1 Introduction

The *tier-based strictly local* (TSL) languages are a restrictive class of subregular languages over strings or trees which model a wide range of long-distance linguistic dependencies, from consonant and vowel harmony to movement and case licensing (cf. Heinz, 2018; Graf, 2022a). Elements which are irrelevant to a given dependency are treated as invisible, and those remaining are treated as adjacent, forming a structure called a *tier*. From this perspective, a syntactic or phonotactic grammar consists of many intersecting TSL patterns with different tiers. For syntax, these include tiers for *wh*-movement, EPP-movement, ϕ -agreement, etc., plus a tier including all elements to regulate local dependencies.

Generally speaking, linguistic dependencies are subject to various blocking effects, including locality restrictions such as the lack of raising out of finite clauses in English (known as *hyperraising*) as well as the well-known island constraints (see Belletti 2018 for an overview). Exactly which elements block which dependencies varies somewhat across languages, though there are some general tendencies (Keine, 2019). Roughly speaking, it is

assumed in the TSL model that dependent elements must be adjacent on a tier; if any other elements intervene on the tier then blocking effects result. Thus, variation in blockers across languages and phenomena equates to differences in the relevant set of tier elements. For example, all C heads appear on the EPP-movement tier in English, but not in a language which allows hyperraising (Graf, 2022b).

While this parameter of the model allows good empirical coverage, it also presents a learning difficulty due to the large number of logically possible tiers, which grows exponentially with the number of elements (segments or syntactic heads). There exist efficient algorithms for learning TSL string patterns, but they either require the tier elements to be fixed in advance (Lambert et al., 2021) or they are not robust against interaction with other constraints (Jardine and McMullin, 2017; Lambert, 2021). The problem is particularly acute for syntax, for even if we can reduce the problem to learning of TSL string languages, the number of tiers and the size of a syntactic lexicon make exhaustive search completely impractical.

A solution may be found by looking to empirically-motivated models of child language acquisition. Heuser et al. (2024) present a model for learning island constraints which constructs a grammar of local bigrams from attested movement paths, supplemented by generalization by the Tolerance Principle (Yang, 2016). They also show that this model makes correct generalizations based on a realistic input distribution. This approach is interesting in that it circumvents the difficulties of tier detection, but only because it lacks tiers altogether: the resulting grammar is *strictly local* (SL) rather than TSL. This brings several limitations, particularly that it can only recognize movement paths which have been delimited in advance.

Ideally, we would like to combine the generality and typological success of the TSL model with an efficient, linguistically-motivated learning algo-

rithm such as that in Heuser et al. (2024). Towards this end, I adapt their algorithm to produce TSL grammars as used in subregular syntax. I also draw attention to several linguistically interesting aspects of the model, which derives a version of the Height-Locality Connection—the observation that higher categories in the clausal spine are subject to fewer locality restrictions—similar to that given in Keine (2019). It is also equally applicable to other pairwise dependencies such as agreement.

The remainder of this paper is laid out as follows. §2 presents a model of syntactic dependencies based on *ancestor strings* (Shafiei and Graf, 2020), whose grammars will be our learning target. §3 adapts the algorithm from Heuser et al. (2024) to the subregular framework, and §4 modifies it to produce a TSL grammar. §5 shows how this model derives a version of the Height-Locality Connection. §6 concludes.

2 Subregular syntax with ancestor strings

This section introduces the class of TSL string languages along with a model of syntactic dependencies based on *ancestor strings* (a-strings, Shafiei and Graf 2020). We begin with the more restrictive class of *strictly local* or SL languages, which model local linguistic dependencies, before moving on to the TSL languages. Examples of string-like constraints from syntax are provided. From there, we discuss the syntactic framework which provides the relevant strings, and the limits of this model.

2.1 Strictly local languages

Many classes of subregular languages, including SL and TSL, are defined in terms of k -factors, which for these classes are substrings, i.e. discrete k -grams. The definitions here follow Mayer (2021).

Let Σ be a fixed alphabet, let s be a string over Σ^* , and let $\bowtie, \bowtie \notin \Sigma$ be the left and right edge markers. The set $f_k(s)$, the k -factors of s , consists of all the length- k substrings of $\bowtie^{k-1}s\bowtie^{k-1}$ where $k \geq 1$. For example, $f_2(ababac) = \{\bowtie a, ab, ba, ac, c\bowtie\}$.

An SL grammar is just a set of forbidden k -factors of fixed width, and its language consists of all strings which do not contain any of these k -factors. Formally:

Definition 1 A *strictly k -local* (SL- k) grammar is a set $G \subseteq (\Sigma \cup \{\bowtie, \bowtie\})^k$. A language $L \subseteq \Sigma^*$ is SL- k iff there exists an SL- k grammar G such that $L = \{s \in \Sigma^* : f_k(s) \cap G = \emptyset\}$.

Alternatively, an SL- k grammar can be defined in terms of permitted k -factors. A set of forbidden factors is a *negative* grammar; its complement, the set of permitted factors, is a *positive* grammar. There are circumstances where either form may be more convenient. When necessary, these will be disambiguated using a superscript: G^+ for a positive grammar and G^- for a negative grammar.

Example 1 Consider the hierarchy of functional categories in a typical English clause. In the sentence *The pizza has been eaten*, it consists of the sequence of categories $T \cdot \text{Perf} \cdot \text{Prog} \cdot v$. Let us assume that the general form of the hierarchy is

$$T > (\text{Perf}) > (\text{Prog}) > (\text{Pass}) > v$$

where categories in parentheses are optional.

The set of licit sequences in a functional hierarchy can be encoded using an SL-2 grammar. Though modeled as a string, in the syntactic framework to be developed in §2.3, it represents a path through part of the tree. The positive grammar is as follows (ignoring edge markers for simplicity):

$$G^+ = \left\{ \begin{array}{l} T \text{ Perf}, \\ T \text{ Prog}, \quad \text{Perf Prog}, \\ T \text{ Pass}, \quad \text{Perf Pass}, \quad \text{Prog Pass}, \\ T v, \quad \text{Perf } v, \quad \text{Prog } v, \quad \text{Pass } v \end{array} \right\}$$

The corresponding negative grammar is:

$$G^- = \left\{ \begin{array}{l} T T, \text{ Perf T}, \quad \text{Prog T}, \quad \text{Pass T}, \quad v T, \\ \text{Perf Perf}, \text{ Prog Perf}, \text{ Pass Perf}, \quad v \text{ Perf}, \\ \text{Prog Prog}, \text{ Pass Prog}, \quad v \text{ Prog}, \\ \text{Pass Pass}, \quad v \text{ Pass}, \\ v v \end{array} \right\}$$

Every 2-factor in our example string appears only in the positive grammar.¹ \lrcorner

2.2 Tier-based strictly local languages

A TSL language is similar to an SL language except that certain symbols are ignored. Let $T \subseteq \Sigma$ be a *tier alphabet*. The string $\pi_T(s)$ is the *tier projection* of s , the result of deleting all σ in s such that $\sigma \notin T$, and concatenating those that remain. For example, if $\Sigma = \{x, a, b, c\}$ and $T = \{a, b, c\}$ then $\pi_T(axxbxxc) = \pi_T(xxxabcxxx) = abc$.

Definition 2 A *tier-based strictly k -local* (TSL- k) grammar is a tuple (T, G) , where T is a tier alphabet and G is an SL- k grammar over T . A language $L \subseteq \Sigma^*$ is TSL- k iff there exists a TSL- k grammar such that $L = \{s \in \Sigma^* : f_k(\pi_T(s)) \cap G = \emptyset\}$.

¹It is not necessary for every functional head to always be present. If syntax includes SL computations then it can implement functional hierarchies just as easily as category selection. See Hanson (2023) for details.

By definition, all symbols not in T may be freely inserted and deleted without affecting the well-formedness of a given string w.r.t. a given TSL grammar, a fact that will be important to the discussion of tier identification in §4.2.

Example 2 DP subjects in English are thought to move to Spec-TP, whether from inside vP or an embedded non-finite TP (the raising construction); they cannot move from a finite CP (hyperraising). Examples are given in (1) below. This dependency—call it EPP-movement—can be encoded with a TSL-2 grammar which requires the mover and landing site (marked with an “EPP” subscript) to be adjacent on a tier. In anticipation of the syntactic framework to be developed, we model this dependency with a string which encodes each head along the movement path, projecting a tier that contains only the relevant elements (movers, landing sites, and blockers).²

- (1) a. We [$_{VP}$ ___ have a problem].
 Path: $\times \cdot D_{EPP} \cdot v \cdot T_{EPP} \cdot \times$
 Tier: $\times \cdot D_{EPP} \cdot T_{EPP} \cdot \times$
- b. We seem [$_{TP}$ to ___ have a problem].
 Path: $\times \cdot D_{EPP} \cdot v \cdot T \cdot V \cdot v \cdot T_{EPP} \cdot \times$
 Tier: $\times \cdot D_{EPP} \cdot T_{EPP} \cdot \times$
- c. *We seem [$_{CP}$ **that** ___ have a problem].
 Path: $\times \cdot D_{EPP} \cdot v \cdot T \cdot C \cdot V \cdot v \cdot T_{EPP} \cdot \times$
 Tier: $\times \cdot D_{EPP} \cdot C \cdot T_{EPP} \cdot \times$

D_{EPP} and T_{EPP} are adjacent on the tier in the licit examples (tier $\times \cdot D_{EPP} \cdot T_{EPP} \cdot \times$) but not in the hyperraising example (tier $\times \cdot D_{EPP} \cdot C \cdot T_{EPP} \cdot \times$). As will be discussed shortly, we aim only to ensure that the mover is immediately followed by the landing site. Accordingly, we only need to ban substrings which consist of a mover followed by anything else. Thus, we have the following grammar:

- (2) Grammar for EPP-movement
- $$T = \{ D_{EPP}, T_{EPP}, C \}$$
- $$G^- = \{ D_{EPP} \cdot D_{EPP}, D_{EPP} \cdot C, D_{EPP} \cdot \times \}$$

The reader may confirm that the tier for the hyperraising example contains the illicit 2-factor $D_{EPP} \cdot C$, while the tiers for the grammatical examples contains no illicit 2-factors. \lrcorner

Essentially, a TSL grammar allows us to ignore elements like VP, NP, etc., which are irrelevant to the long-distance dependency in question. The next subsection introduces a syntactic framework which provides the strings assumed in the above examples.

²Following MG convention, intermediate landing sites are not modeled directly.

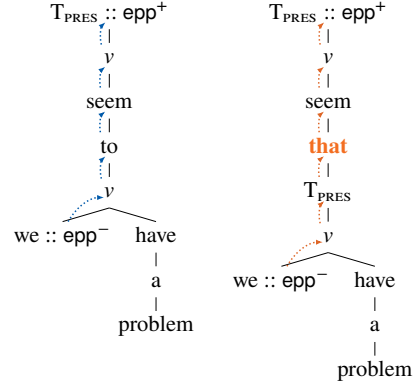


Figure 1: Dependency trees for *We seem to have a problem* (left) and **We seem that have a problem* (right) showing a-strings for moving elements. In the latter structure, *that* intervenes, preventing movement.

2.3 Dependency trees and ancestor strings

Following recent work in subregular syntax (Shafiei and Graf, 2020; Graf, 2022b, a.o.), I use MG dependency trees for the syntactic representation. Examples for sentences (1b) and (1c) are given in Figure 1. In these trees each node is a lexical item; compared to X-bar trees, each head and its projections are collapsed into a single node. The daughters of a node are its arguments, ordered from right to left in order of first merge, such that the rightmost daughter is the complement and all others are specifiers. For example, the right daughter of embedded v is the head of the complement VP, and the left daughter is the head of the DP subject (its specifier). In addition, each node is annotated with MG features guiding the Merge and Move operations (cf. Stabler, 1997, 2011). Since we are not concerned with local dependencies here, only Move features are shown. Positive features mark landing sites, and negative features mark moving elements. For example, finite T bears epp^+ and the subject D head bears epp^- . Note that all elements appear in their base positions only, as in standard MG derivation trees.

Let us now implement a string-based model of movement constraints in which we extract the path from each mover to the root of the tree. Essentially, we take the order imposed by the (inverted) dominance relation and ignore the sibling relation. Shafiei and Graf (2020) call such paths *ancestor strings*, or a-strings, which they used to model a subset of the island constraints, including the *wh*-island constraint and the complex NP island constraint. First, we will see how this works for EPP-movement, then briefly discuss *wh*-movement.

Example 3 In order to keep the notation concise, I substitute most lexical items with their categories, and place the movement features as subscripts without the +/- diacritic, as before. Thus, the a-strings for the EPP movers in the structures in Figure 1 are:

$$\begin{aligned} \text{Raising } (\checkmark): & \quad D_{\text{EPP}} \cdot v \cdot T \cdot V \cdot v \cdot T_{\text{EPP}} \\ \text{Hyperraising } (\times): & \quad D_{\text{EPP}} \cdot v \cdot T \cdot C \cdot V \cdot v \cdot T_{\text{EPP}} \end{aligned}$$

These are exactly the same strings as before, so we can continue to use the grammar in (2). \lrcorner

Example 4 The *wh*-island constraint can be described as a ban on *A'*-movement paths (including but not limited to *wh*-movement) which are interrupted by an interrogative CP, as illustrated by the difference between (3a) and (3b). Movement paths (a-strings) and their *wh*-tiers are included below each example, and the full structures are shown in Figure 2. For simplicity, we abstract away from EPP-movement and model only *wh*-movement.

- (3) a. What did you think **that** John ate ___?
 Path: $D_{\text{WH}} \cdot V \cdot v \cdot T \cdot \text{that} \cdot V \cdot v \cdot T \cdot C_{\text{WH}}$
 Tier: $D_{\text{WH}} \cdot C_{\text{WH}}$
- b. *What did you wonder **whether** John ate ___?
 Path: $D_{\text{WH}} \cdot V \cdot v \cdot T \cdot \text{whether} \cdot V \cdot v \cdot T \cdot C_{\text{WH}}$
 Tier: $D_{\text{WH}} \cdot \text{whether} \cdot C_{\text{WH}}$

We can construct a very similar grammar to the previous one which captures this blocking effect:

- (4) Grammar for *wh*-island constraint
 $T = \{D_{\text{WH}}, C_{\text{WH}}, \text{whether}\}$
 $G^- = \{D_{\text{WH}} \cdot D_{\text{WH}}, D_{\text{WH}} \cdot \text{whether}, D_{\text{WH}} \cdot \times\}$

As before, the tier projection for the island violation contains the illicit 2-factor $D_{\text{WH}} \cdot \text{whether}$, while the non-island structure contains no such 2-factors. \lrcorner

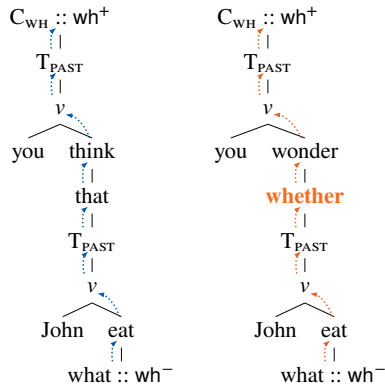


Figure 2: Dependency trees for *What did you think that John ate?* (left) and **What did you wonder whether John ate?* (right). In the latter structure, *whether* intervenes.

Note that because the a-string of a node extends to the root of the tree, it may contain fragments of other movement paths as well as nodes that are not part of any movement path. Our grammar is constructed in such a way that this does not pose an issue. However, the approach does have several limitations, as discussed below. Additionally, applying Heuser et al.’s algorithm to a-strings requires this extra material to be removed, as discussed in §3.

Also note that TSL grammars such as those in (2) and (4) enforce only these constraints and no others. As alluded to in the introduction, we must intersect these and other constraints, including local constraints, to produce a *multi-TSL* (MTSL) grammar. This is just a set of pairs of tier alphabets and associated constraints (grammars with the same tier alphabet can be intersected directly); see De Santo and Graf (2019) for details.

2.4 The strengths and limitations of a-strings

A-strings encode only enough information to enforce constraints base on containment (dominance) from the perspective of the mover. Shafiei and Graf (2020) use them to model island constraints, and as we have seen, certain other blockers can be handled in the same manner. We can also ensure that the mover has a landing site and capture some cases of relativized minimality, namely those where a mover contains another mover of the same type.

So, what can a-strings not do? Notably, they do not allow us to ensure that every landing site has exactly one mover. This requires tree tiers, as in Graf (2022b). They also cannot handle all cases of relativized minimality, as c-commanding specifiers do not appear in an a-string; this requires the *command strings* (c-strings) of Graf and Shafiei (2019). Additionally, to model specifier islands, information encoding left branches must be added to the string. See Shafiei and Graf (2020) for further discussion. The focus of this paper is on learning the tier alphabet; for this the a-string model will suffice, and the results should in principle extended to more complete models.

3 Distributional learning of syntactic blockers

I now describe the algorithm from Heuser et al. (2024), adapted to the syntactic framework presented in the previous section. We then discuss the ways in which the algorithm can do more than it was originally intended to, but being essentially

an SL learner rather than a TSL learner, is not a complete solution on its own.

3.1 Preliminaries

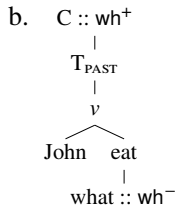
The algorithm assumes that the learner has already parsed the input and identified both moved elements and their initial positions. Now, they must determine the licit paths from the mover to the landing site for each type of movement. This can be cast as the learning an SL-2 grammar over (truncated) a-strings for each movement dependency. It is also assumed that the learner will generalize to unseen paths via the Tolerance Principle (TP, Yang 2016). The equates to a procedure for adding some but not all missing k -factors to the grammar.

While some readers may worry about taking the tree structure as a given, this essentially reduces to the assumption that long-distance syntactic dependencies are parasitic on local constituent structure, which must be learned regardless. Similarly, some other mechanism is responsible for identifying moved elements. It is conceivable that each of these can be learned distributionally with the TP, though such work is still in its infancy. See, e.g., Liang et al. (2022) regarding the learning of syntactic categories, and Li and Schuler (2023) regarding recursive embedding.

3.2 Tracking bigrams

Consider the *wh*-object question in (5), assumed to be in the input. The learner gathers from this that $\text{what} \cdot \text{eat} \cdot v \cdot T_{\text{PAST}} \cdot C_{\text{WH}}$ is a licit movement path, but does not know (yet) that every sequence of categories $D_{\text{WH}} \cdot V \cdot v \cdot T \cdot C_{\text{WH}}$ is a licit path.

(5) a. What did John eat ____?



c. a-string: $\text{what} \cdot \text{eat} \cdot v \cdot T_{\text{PAST}} \cdot C_{\text{WH}}$

The learner begins memorizing the attested 2-factors from each path, which is just the procedure for learning a (positive) SL-2 string grammar (Heinz, 2010). From the current example, they learn that $\{\text{what} \cdot \text{eat}, \text{eat} \cdot v, v \cdot T_{\text{PAST}}, T_{\text{PAST}} \cdot C_{\text{WH}}\}$ are all licit 2-factors.³ Heuser et al. show that because functional categories like v and T are few

³I continue to ignore edge markers for simplicity.

in number and frequent in the input, the learner will discover that all combinations may occur. For example, they will learn that *wh*-movement may occur over transitive and intransitive v , past and present tense, and so on.

Note that we must truncate of the a-string at the landing site when it is not the root, since the portion beyond the landing site may contain bigrams which cannot occur along the movement path. For example, in the sentence *Who wonders what John ate?*, the full a-string for *what* contains the bigram $C_{\text{WH}} \cdot \text{wonder}$. If not excluded, the learner would conclude that movement over interrogative C is permitted. We will return to this issue in §3.5.

3.3 Generalizing with the Tolerance Principle

With regard to lexical categories such as verbs, the learner needs to invoke the TP. Given a class of N items and a proposed generalization, the TP states that the learner will adopt the generalization iff the number of items M in this class which are known to fit the generalization exceeds a threshold θ_N , where

$$\theta_N = N/\ln(N)$$

In this case, N is the total number of verbs they have learned, and M is the number that have been attested with *wh*-movement. Heuser et al. show that for English, *wh*-movement of objects occurs with a large proportion of the most frequent verbs in child-directed speech—the number of exceptions far below the threshold—so the learner will adopt the generalization that *wh*-movement is permitted across all verbs. This is equivalent to adding all missing 2-factors of the form $D_{\text{WH}} \cdot V$ and $V \cdot v$ to the grammar.⁴

This brings us to islands. Once the learner observes cross-clausal movement from an embedded declarative such as (3a), they will add $T_{\text{PAST}} \cdot \text{that}$ and $\text{that} \cdot \text{think}$ to the grammar. But if movement across a certain structure, such as the *wh*-island violation in (3b), is not attested, and the TP does not permit generalization, then the relevant 2-factors will never be added to the grammar. Heuser et al. (2024) show that this is indeed what we expect for “strong islands” in English. They also show how this derives the fact that not all verbs which take CP complements allow *wh*-movement, forming so-called “selective islands”. Although the learner

⁴The TP does not provide the class of possible generalizations, only whether a given generalization is “good enough”. For present purposes, I assume that syntactic categories such as $V/A/N/P/T/C$ are the only conditioning factors.

observes *wh*-movement across verbs like *think* and *say*, they do not observe movement across verbs such as *complain* and *quip*, and there are too many such verbs for the TP to permit generalization to the full class of verbs which select for a CP.

3.4 Beyond islands

To briefly summarize, the algorithm constructs a positive SL-2 grammar encapsulating the crucial information about licit and illicit movement paths where blockers are effectively encoded as missing 2-factors. Although not discussed by Heuser et al. (2024), the approach is equally applicable to other restrictions on movement such as those discussed by Keine (2019), which are the focus of §5.

It is also applicable to non-movement dependencies, to the extent that pairs of dependent items can be identified. Shafiei and Graf (2020) note that constraints on long-distance linguistic patterns tend to take involve a domain and blockers within that domain. For movement, the domain elements are movers and their landing sites, while for agreement we have, in Minimalist terms, elements with unvalued and valued features of the same type. Indeed, Keine’s version of the Height-Locality Connection treats movement and agreement equally. If these are learned in the same way, then we have an explanation for this close correspondence.

Finally, note that the same properties that allow learning of weak islands also allow for cross-linguistic variation such as the availability of hyper-raising (Charles Yang, p.c.). Specifically, it predicts that hyperraising should only be allowed if robustly attested in the input. This, of course, raises the question of how such structures ever arise. But we could just as easily ask the same of long-distance *wh*-movement, which is by now known to be more or less restrictive in different languages. For now, we must set these diachronic questions aside.

3.5 Limitations of SL learning

The fact that the Heuser et al. (2024) algorithm is essentially an SL learner means that the resulting grammars cannot be applied to arbitrary *a*-strings, only those which start with a mover and which are truncated at the first landing site. This is because it is in general not possible for an SL grammar to relate two elements which do not occur in the same *k*-factor. As a consequence, it is impossible to ensure that there is exactly one landing site per mover, nor to detect whether a blocker actually occurred along a movement path and not somewhere else. In

contrast, our TSL grammars from §2 do not suffer from either restriction.

Thus, truncating the *a*-string only creates the illusion that SL is adequate. While this operation is useful in the learning algorithm, including it in the grammar would increase its power, producing a class that is quite different from TSL.⁵ Instead, what we want to do is to take the information that was obtained using this technique and encode it in a TSL grammar, which has the right formal properties. This is the topic of the next section.

4 Constructing the tier

To review the discussion so far, we can frame our learning problem as follows: given a corpus of MG dependency trees, how do we discover the TSL constraints on long-distance dependencies over *a*-strings? In particular, how do we discover which elements other than the dependent items are visible?

We have already seen how Heuser et al.’s path-based algorithm forms the foundation of an appealing solution, but on its own is not enough. This section begins with a more detailed summary of the issues involved with TSL learning before attempting to bridge the gap by modifying the Heuser et al. algorithm to produce a TSL grammar.

4.1 The problem of learning tiers

TSL languages are efficiently learnable given a fixed tier alphabet and *k*-factor size (Lambert et al., 2021), but this may not be a realistic assumption for natural language. There is reason to think that the value of *k* rarely exceeds 2 for long-distance constraints (McMullin, 2016; Graf, 2022b; Hanson, 2024), but it is far less clear that the tier alphabet can be known in advance. Because the number of possible tiers alphabets is exponential in the size of the full alphabet (it is $2^{|\Sigma|}$), we must avoid exhaustive search of this space. While there exist efficient (polynomial time) algorithms that determine the tier alphabet from positive data (Jardine and McMullin, 2017; Lambert, 2021), these are not robust against interaction with other constraints. Since natural language almost always involves the interaction of many constraints, this prevents such algorithms from being used with real world data.

One way of tackling the problem is to find ways to pare down the hypothesis space such that the brute force method becomes practical. For example, we

⁵It would be a subclass of IBSP. Shafiei and Graf (2020) also use IBSP, although in a very different manner.

could appeal to formal universals on the relations between the alphabets of different tiers (Aksënova and Deshmukh, 2018). Alternatively, we could make use of substantive universals such as some version of the Height-Locality Connection; Keine’s (2019) version says that a “lower” category can be a blocker for a “higher category”, e.g. v cannot be a blocker for a landing site at T.

Another possibility, which I pursue here, is to identify a set of heuristics which allows the learner to discover the tier alphabet without ever engaging in exhaustive search. In other words, the supposedly impossible tiers are in fact perfectly valid, but the learner will never posit them under normal conditions due to the way in which they navigate the hypothesis space. The crucial heuristic in this case, taken from Heuser et al. (2024), is that by restricting our attention to the path between two dependent elements, we can identify its blockers, which must appear on the same tier.

In this case, the Height-Locality Connection becomes a side effect of the learning process rather than a cause, and is also unified with the theory of islands. As discussed earlier, the close similarity of movement and agreement constraints is derived as well. Yet another issue with existing TSL learners is that they all involve exact identification in the limit, whereas children must generalize from limited data. Though orthogonal to our main focus, the adoption of the TP largely solves this problem as well. Altogether, the proposed approach not only solves several major learnability problems for the TSL model, but also adds several typological predictions which are not inherent to the model.

4.2 From local to tier-based constraints

Existing TSL learners infer the tier alphabet by utilizing a definitional property of a TSL- k language: any symbol not on the tier can be freely inserted and deleted without changing the well-formedness of a string. As discussed by Lambert (2021), we can do this by keeping track of just the sets of attested local k -factors and $(k+1)$ -factors. Since the k -factors can themselves be obtained from the $(k+1)$ -factors, only the latter must be memorized. Thus, in principle we can use the local 2-factors discovered by Heuser et al. (2024)’s algorithm to identify tier-based 1-factors, which are the blockers themselves. By recombining these blockers with the dependent items that bookend the path, we can construct the desired TSL-2 grammar.

However, we have still not addressed the problem

of interaction with local constraints. Detecting free insertion and deletion as described above requires collecting every possible local $(k+1)$ -factor in a TSL language, but the existence of other constraints means that this will never happen. For instance, every permutation of every subset of a functional hierarchy would have to occur in the input for these elements to be removed from the tier.

I propose that we can solve this problem by using the background grammar encoding local constraints as the standard of comparison for free insertion and deletion. Recall the behavior of our path-based learner for *wh*-movement structures such as those in (3a) and (5). After decomposing paths and applying the TP, the resulting grammar will contain a dense network of 2-factors of the form $\{D_{WH} \cdot V, V \cdot v, v \cdot T, T \cdot C_{DECL}, C_{DECL} \cdot V\}$, but not $T \cdot$ whether or whether $\cdot V$. All of these 2-factors are licit when they do *not* occur along a *wh*-movement path, and are therefore part of the local constraint grammar. As a result, we can infer that *whether* is a blocker due to the conspicuous absence of 2-factors which contain it. In contrast, 2-factors like $T \cdot v$ (reverse order) and $V \cdot C$ (skipping T) are already missing in the local constraint grammar, so their absence in the movement path grammar can be ignored.

4.3 Algorithm

The proposed algorithm is as follows. Let G_L^2 be the positive SL-2 grammar for local constraints and G_M^2 be the grammar for movement type M . Construct G_L^2 by collecting all 2-factors from all a-strings, and construct G_M^2 from truncated a-strings as before. Add missing 2-factors to each where permitted by the TP. Next, construct G_L^1 and G_M^1 by decomposing the 2-factors in G_L^2 and G_M^2 into their constituent 1-factors.

Now we test for tier membership. Free deletion is vacuous for TSL-1, since it is trivially true that for every symbol, removing that symbol from an attested 2-factor which contains it in a certain position produces an attested 1-factor (this not necessarily true for larger values of k).

The crucial test, corresponding to the free insertion test, tests for factors missing from G_M^2 but present in G_L^2 . Let $G_D^2 = G_L^2 \setminus G_M^2$. For every symbol, we ask if it can be added to either side of 1-factor in G_M^1 to produce a 2-factor in G_D^2 ; if so, then the symbol is a blocker. Finally, we construct the target TSL-2 grammar, which consists of 2-factors containing the mover followed by a blocker, another mover, or the right edge marker.

Example 5 Given typical data, the grammar G_M^2 for *wh*-movement will include all 2-factors of the form $\{D_{WH} \cdot V, V \cdot v, v \cdot T, T \cdot C_{DECL}, T \cdot C_{WH}\}$. It also contains *that* · think and *that* · say but not *that* · complain or *that* · quip. G_L^2 contains all of these, so the difference G_D^2 includes *that* · complain and *that* · quip. If we consider the elements *complain* and *quip*, we could add *that* from G_M^1 to 2-factors in G_D^2 , so they are blockers. In contrast, even though *that* has containing 2-factors in G_D^2 , these cannot be constructed by adding a symbol from G_M^1 , so they are not blockers. \lrcorner

Based on examples like these, it would appear that comparing just G_L^1 and G_M^1 is sufficient, since any element in G_L^1 but not G_M^1 is guaranteed to have a containing factor in G_D^2 . If this reasoning is correct, it may be possible to simplify the above procedure. However, it renders the relation to Lambert (2021) opaque, and there may be corner cases which have not been considered. Also, the fact that movement paths are calculated from the base position could affect the predictions of the model when we look beyond EPP-movement and *wh*-movement. I leave the investigation of such details to future work.

4.4 Discussion

The reader may be wondering why we do not simply track local 3-factors in order to directly infer tier-based 2-factors. There are several problems with this method, but first and foremost is that it greatly increases data sparsity. Although Heuser et al. (2024) found empirical success with local 2-factors, it is not clear whether the TP will allow the same generalizations when applied to 3-factors.

Next, I should describe how the model could be extended beyond domain-based constraints on movement. Handling agreement should be straightforward; we just need to add positive and negative agreement features analogous to MG movement features, as in Hanson (2024). Other dependencies such as case assignment would require identification of the relevant domain nodes (i.e. as in dependent case theory), and we could in principle adapt the algorithm to c-strings in order to identify constraints on c-commanding elements.

Finally, I wish to briefly mention some alternative approaches to learning long-distance syntactic dependencies. Many of these are probabilistic models; for example, the model in Pearl and Sprouse (2013) tracks path trigram probabilities in order to learn

syntactic islands. This is not entirely dissimilar to the present model, except that we do not attempt to learn gradient constraints. It is, of course, possible to introduce gradience into subregular models; see Mayer (2021) and Torres et al. (2023). The present paper, by incorporating a TP-based model, relegates the use of frequency/probability to a small corner of the learning algorithm. In principle, we could adapt it to produce a probabilistic TSL grammar by comparing k -factor probabilities rather than discrete k -factors.

5 On the Height-Locality Connection

The Height-Locality Connection (HLC) is the observation that restrictions on long-distance syntactic dependencies correlate with the category of the “height” of the upper element (e.g. landing site) such that higher categories can enter into more distant dependencies (Keine, 2019). While several distinct theories can be found in the literature (Williams, 2002; Abels, 2012, a.o.), the present approach is most directly comparable to Keine’s theory of Probe Horizons, in which each type of probe (i.e. a head that hosts a landing site or unvalued feature) has a *horizon* beyond which no dependencies can be formed. In TSL terms, a horizon is simply a blocker on a tier, and in this sense no different from an island, a bounding node in the binding theory, or any other such element. I show here that the learning algorithm from the previous section predicts a version of the HLC which is similar though not identical to Keine’s.

For Keine, the horizon for each combination of major category and active feature is lexically specified. For example, finite T in English bears some feature (which we have been calling epp^+) which triggers movement of the subject. This probe can see into a non-finite TP, but not a finite CP. Thus, C is a horizon for this dependency in English, but it need not be so in other languages. Keine shows T is a horizon for analogous A-movement in Hindi; in languages with hyperraising neither T nor C is a horizon. As we have discussed, this variation is a core prediction of the TSL model as well.

However, according to Keine, it is not the case that any category is a possible horizon for any probe, only those that are *at least as high* as the category of the probe. This means that a probe on T can never have v or V as a horizon, for example.⁶

⁶I refer the reader to Section 5 of Keine (2019) regarding the derivation of this generalization, which is based on

Restricting our attention to the basic clausal spine, this yields the typology of possible horizons shown in (6). Thus, we might rephrase the HLC as saying that higher categories *must* have a larger locality domain; lower categories may see just as far, but have smaller domains as a tendency.

(6)

Category	Possible Horizons
C	C
T	C, T
<i>v</i>	C, T, <i>v</i>

Let us consider how such a generalization could arise from the learning algorithm outlined here. In the case of EPP-movement, the learner observes movement from Spec-*v*P in simple transitive clauses, and out of VP in the case of unaccusatives and passives. When all is said and done, V and *v* do not appear on the tier, and so are not horizons. If the learner also observes raising out of TP (as in English), T will be removed as well, as will C in a language with hyperraising, but for V and *v* this is all but guaranteed, since DPs in general originate within these phrases. By the same logic, the learner will remove C from the tier for *wh*-movement only if cross-clausal movement is observed (as it is in English), but the observation of *wh*-object movement even in simplex clauses necessarily rules out V, *v*, and T since all are below C.

To be fully explicit, the proposed algorithm predicts the HLC to be a tendency rather than a strict rule in *both* directions: lower categories usually have smaller locality domains, and higher categories usually have larger ones, but exceptions are in principle possible in both directions. Again, in our representative examples of EPP-movement and *wh*-movement the relevant class of movers is able to occur in the complement of VP, the lowest possible position in the clausal spine; invisibility of the entire functional sequence below the probe follows as a result. Thus, to determine whether Keine’s generalization is truly correct, we would need to find a class of mover which originates only in higher positions, that is, one which does not include any DPs. At present, I do not know of a good candidate class of movers to perform this test.

To close this section, I wish to reemphasize the generality of the proposed learning algorithm, which is equally relevant to islands and other kinds of blockers. In his discussion of acquisition, Keine

the assumption that functional projections involve “feature inheritance” of lower categories in the functional sequence.

notes that the implicational hierarchy imposed by his theory provides the learner with a safe way of navigating the space of possible horizons, starting with the assumption that the category of the probe is also the lowest horizons, and removing horizons from the grammar as required by the input. This is correct, and our algorithm works from a similar principle. But Keine’s assumption that projections lower than the probe cannot be horizons is not necessary to achieve this.

6 Conclusion

In this paper, I proposed an algorithm which allows for the creation of TSL grammars from the output of Heuser *et al.*’s path-based algorithm, avoiding the need to search the space of tier alphabets. This approach combines the strengths of their algorithm with those of the TSL model, and derives the Height-Locality Connection as a byproduct of the learning process. While this paper used a-strings and focused on movement, the principle of inferring tier-based constraints via comparison of SL grammars should in principle extend to other TSL models of syntax and other dependencies such as agreement and case. I leave investigation of these to future research.

More broadly, this work represents the start of integration between subregular syntax and acquisition theories based on the TP. I am aware only of one other line of work which involves learning TSL grammars with the TP, which is Belth’s (2023) algorithm for learning long-distance harmony. Since subregular linguistics has consistently shown a great deal of formal similarity across domains, it would be prudent to examine whether Belth’s algorithm can be applied to the problem of learning syntactic dependencies, and vice versa. Formal learnability has long been central to subregular linguistics, but as I hope to have shown, future progress may rely on looking also to theories grounded in the empirical facts of child language acquisition.

Acknowledgments

This work partly was supported by the National Science Foundation under Grant No. BCS-1845344 and by an award from the Institute for Advanced Computational Science at Stony Brook University. I thank Jordan Kodner and Sarah Payne for reviewing an early draft of the paper. I also thank three anonymous reviewers for their detailed comments, which helped to improve the clarity of several key issues.

References

- Klaus Abels. 2012. The Italian left periphery: A view from locality. *Linguistic Inquiry*, 43(1):229–254.
- Alëna Aksënova and Sanket Deshmukh. 2018. [Formal restrictions on multiple tiers](#). In *Proceedings of the Society for Computation in Linguistics 2018*, pages 64–73.
- Adriana Belletti. 2018. Locality in syntax. In *Oxford Research Encyclopedia of Linguistics*. Oxford University Press.
- Caleb Belth. 2023. [Towards a learning-based account of underlying forms: A case study in Turkish](#). In *Proceedings of the Society for Computation in Linguistics 2023*, pages 332–342.
- Aniello De Santo and Thomas Graf. 2019. [Structure sensitive tier projection: Applications and formal properties](#). In *Formal Grammar*, pages 35–50, Berlin, Heidelberg. Springer.
- Thomas Graf. 2022a. [Subregular linguistics: bridging theoretical linguistics and formal grammar](#). *Theoretical Linguistics*, 48(3–4):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.
- Thomas Graf and Nazila Shafiei. 2019. [C-command dependencies as TSL string constraints](#). In *Proceedings of the Society for Computation in Linguistics 2019*, pages 205–215.
- Kenneth Hanson. 2023. Strict locality in syntax. In *Proceedings of CLS 59*.
- Kenneth Hanson. 2024. Tier-based strict locality and the typology of agreement. Ms. Stony Brook University.
- Jeffrey Heinz. 2010. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906, Uppsala, Sweden. Association for Computational Linguistics.
- Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry M. Hyman and Frans Plank, editors, *Phonological Typology*, number 23 in Phonetics and Phonology, pages 126–195. De Gruyter Mouton.
- Annika Heuser, Hector Vazquez Martinez, and Charles Yang. 2024. The learnability of syntactic islands. Presentation at NELS 54.
- Adam Jardine and Kevin McMullin. 2017. Efficient learning of tier-based strictly k-local languages. In *Language and Automata Theory and Applications*, pages 64–76, Cham. Springer International Publishing.
- Stefan Keine. 2019. [Selective opacity](#). *Linguistic Inquiry*, 50(1):13–62.
- Dakotah Lambert. 2021. [Grammar interpretations and learning TSL online](#). In *Proceedings of the Fifteenth International Conference on Grammatical Inference*, volume 153 of *Proceedings of Machine Learning Research*, pages 81–91. PMLR.
- Dakotah Lambert, Jonathan Rawski, and Jeffrey Heinz. 2021. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling*, 9(1):151–194.
- Daoxin Li and Kathryn D. Schuler. 2023. Acquiring recursive structures through distributional learning. In *BUCLD 47: Proceedings of the 47th Annual Boston University Conference on Language Development*.
- Kevin Liang, Diana Marsala, and Charles Yang. 2022. Distributional learning of syntactic categories. In *BUCLD 46: Proceedings of the 46th annual Boston University Conference on Language Development*.
- 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.
- Kevin McMullin. 2016. *Tier-based locality in long-distance phonotactics: learnability and typology*. Ph.D. thesis, University of British Columbia.
- Lisa Pearl and Jon Sprouse. 2013. Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem. *Language Acquisition*, 20(1):23–68.
- Nazila Shafiei and Thomas Graf. 2020. [The subregular complexity of syntactic islands](#). In *Proceedings of the Society for Computation in Linguistics 2020*, pages 421–430.
- Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retore, editor, *Logical Aspects of Computational Linguistics*. Springer.
- Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press.
- Charles Torres, Kenneth Hanson, Thomas Graf, and Connor Mayer. 2023. [Modeling island effects with probabilistic tier-based strictly local grammars over trees](#). In *Proceedings of the Society for Computation in Linguistics 2023*, pages 155–164.
- Edwin Williams. 2002. *Representation theory*. MIT Press.
- Charles Yang. 2016. *The price of linguistic productivity: How children learn to break the rules of language*. MIT Press.

Correlation Does Not Imply Compensation: Complexity and Irregularity in the Lexicon

Amanda Doucette¹ Ryan Cotterell² Morgan Sonderegger¹ Timothy J. O'Donnell^{1,3}

¹Dept. of Linguistics, McGill University ²Dept. of Computer Science, ETH Zurich

³Canada CIFAR AI Chair, Mila

amanda.doucette@mail.mcgill.ca ryan.cotterell@inf.ethz.ch
morgan.sonderegger@mcgill.ca timothy.odonnell@mcgill.ca

Abstract

It has been claimed that within a language, morphologically irregular words are more likely to be phonotactically simple and morphologically regular words are more likely to be phonotactically complex. This inverse correlation has been demonstrated in English for a small sample of words, but has yet to be shown for a larger sample of languages. Furthermore, frequency and word length are known to influence both phonotactic complexity and morphological irregularity, and they may be confounding factors in this relationship. Therefore, we examine the relationships between all pairs of these four variables both to assess the robustness of previous findings using improved methodology and as a step towards understanding the underlying causal relationship. Using information-theoretic measures of phonotactic complexity and morphological irregularity (Pimentel et al., 2020; Wu et al., 2019) on 25 languages from UniMorph, we find that there is evidence of a *positive* relationship between morphological irregularity and phonotactic complexity within languages on average, although the direction varies within individual languages. We also find weak evidence of a negative relationship between word length and morphological irregularity that had not been previously identified, and that some existing findings about the relationships between these four variables are not as robust as previously thought.¹

1 Introduction

The *compensation hypothesis* (Martinet, 1955; Hockett, 1955) states that as a language increases in complexity in one area, another must decrease in complexity to compensate. A compensatory relationship could exist either *within* a language (i.e., words that are more complex in one way are less complex in another), or *across* languages (i.e., an entire lexicon that is more complex in one way

is less complex in another). One such compensatory relationship has been proposed between morphological irregularity and phonotactic complexity. Hay (2003), Hay and Baayen (2003), and Burzio (2002) argue that words within a language with irregular morphology tend to be phonotactically simple, while words with regular morphology tend to be phonotactically complex.

Although there is some evidence for this relationship in English (reviewed below), the existence of a correlation does not imply *compensation*, which we take to mean that an increase in one variable directly causes a decrease in the other (Pearl et al., 2016, §1.5). While we may observe a correlation between morphological irregularity and phonotactic complexity, it is possible that there is in fact no direct causal relationship between them. For example, they could share a common cause such as word frequency (Pearl et al., 2016, §2.2). The effect could also be mediated through a third mediator variable (Pearl et al., 2016, §3.7), as has been argued for the relationship between phonotactic complexity and frequency (Mahowald et al., 2018).

Therefore, to assess the relationship between morphological irregularity and phonotactic complexity, we need to examine any other variables they may be related to. Previous work suggests that both morphological irregularity and phonotactic complexity are correlated with word frequency. Wu et al. (2019) showed that morphological irregularity positively correlates with frequency, and Mahowald et al. (2018) showed that phonotactic complexity inversely correlates with frequency after controlling for word length. Phonotactic complexity is also known to be correlated with word length, with longer words conveying less information per phoneme (Pimentel et al., 2020), and more frequent words tend to be shorter (Zipf, 1935; Piantadosi et al., 2011; Piantadosi, 2014; Pimentel et al., 2023).

While there is evidence supporting a relation-

¹Code is available at <https://osf.io/ax78p/>.

ship between some pairs of these four variables—morphological irregularity, phonotactic complexity, word length, and frequency—there is reason to be uncertain about the existence and direction of a correlation between others, whether the correlation holds within or across languages, and what other variables need to be controlled for to accurately assess the effect. The pairwise relationships between these variables have not yet been examined on a single data set of many languages, and some relationships have only been examined using orthographic rather than phonetic transcriptions. Therefore, in addition to examining the relationship between phonotactic complexity and morphological irregularity, we will also examine the relationships between all other pairs of variables in this set.

We find that within languages, there is a positive effect of phonotactic complexity on morphological irregularity after controlling for word length and frequency. Across languages, we find no consistent effect. We replicate previous findings of a negative effect of word length on frequency, and of a positive effect of frequency on morphological irregularity. We also find a negative effect of frequency on phonotactic complexity, although not as robust as previously suggested. Our results for the relationship between phonotactic complexity and word length complicate previous results: We find a positive effect for one data set and a negative effect for another. Finally, we present a novel analysis of the effect of word length on morphological irregularity, and find a negative effect in most languages.

2 Background

For each pair of variables, we summarize previous work demonstrating a correlation, any theoretical arguments supporting a positive, negative, or no relationship between the two, and what other variables must be controlled for to examine a potential causal relationship.

2.1 Phonotactic Complexity vs. Morphological Irregularity

It has been hypothesized that within a language, phonotactic complexity is negatively correlated with morphological irregularity. Hay and Baayen (2003), Hay (2003), and Burzio (2002) argued that for English, words that are phonotactically complex are more likely to be morphologically regular. For example, *dreamed* is morphologically regular, but contains the unusual consonant cluster [md], while

went is morphologically irregular, but has regular phonotactics. While this relationship has not been examined in other languages, there are several reasons to suspect a negative correlation as a universal tendency. First, low-probability phonotactic junctures can facilitate morphological decomposition, as argued by Hay (2003), who found that for a set of 12 English affixes the proportion of words creating an illegal phonotactic juncture was predictive of morphological productivity. Second, as argued by Burzio (2002), irregular forms are more likely to be memorized, while regular forms are constructed from individual morphemes. If phonotactically simple words are easier to store in memory, phonotactic complexity should be inversely correlated with morphological irregularity.

There are also reasons to suspect no relationship between morphological irregularity and phonotactic complexity, related to the limitation that all previous work considers only a small set of words. Morphological and phonotactic processes could apply independently, and previously observed significant correlations could be statistical accidents due to small sample size. Indeed, responding to Hay (2003), Plag (2002) found no correlation between morphological irregularity and phonotactic complexity in a different sample of 12 English affixes. Alternatively, morphological irregularity and phonotactic complexity could be independent conditional on a third common cause or mediator variable, with which they are both correlated. This could result in a statistically significant correlation, while there is no causal relationship in reality.

One such common cause that could result in a positive observed correlation between phonotactic complexity and morphological regularity is word age. Hay and Baayen (2003) note that highly productive affixes are regularly used in creating new words. Thus, new words in a language will tend to have regular morphology, and it seems plausible they will also tend to have regular phonotactics. As the language changes over time, what is considered regular will also change, resulting in a positive correlation: Older words will have irregular morphology and high phonotactic complexity, while newer words will have regular morphology and low phonotactic complexity. Other possible common causes include word length and frequency: Previous work demonstrates correlations between frequency and phonotactic complexity, frequency and morphological irregularity, and word length and phonotactic complexity. A negative effect of word

length on morphological irregularity is also plausible, and will be established in our data. Both word frequency and word length are therefore common causes that should be controlled for in assessing the relationship between phonotactic complexity and morphological irregularity.

2.2 Phonotactic Complexity vs. Length

Pimentel et al. (2020) demonstrated a strong negative correlation between phonotactic complexity and average word length both across and within 106 languages. Pellegrino et al. (2011) suggest that this compensation is the result of a linguistic universal: The rate of information in every language is very similar, with the amount of information per word roughly constant. Thus, longer words should have less information per phoneme (Coupé et al., 2019; Meister et al., 2021). While previous work (discussed below) suggests that frequency is a common cause of both phonotactic complexity and word length, and should be controlled for in this analysis, we will not control for it in line with our goal of replicating previous studies using a single dataset.²

2.3 Morphological Irregularity vs. Frequency

A positive correlation between morphological irregularity and word frequency has been observed in English (Marcus et al., 1992; Bybee, 1985), but this correlation was questioned by Fratini et al. (2014) and Yang (2016). In a larger set of 21 languages, Wu et al. (2019) found a positive correlation between morphological irregularity and frequency. These correlations were found to be more robust when irregularity was considered as a property of lemmas rather than individual words. A potential mechanism is described by Hay and Baayen (2003): More frequent words are more likely to be accessed as whole words, and less frequent words are more likely to be parsed into their component morphemes. Because irregulars are more likely to be accessed as whole words in memory, there will be a positive correlation between frequency and morphological irregularity. In contrast, if we assume that lexicons are optimized for efficient communication, i.e., more frequent words should be less morphologically complex (Zipf, 1935), we would expect frequent words to have regular morphology, i.e., a negative correlation.

²However, preliminary models show that controlling for frequency only has a minimal impact on results.

2.4 Phonotactic Complexity vs. Frequency

A consequence of Zipf's (1935) hypothesis that the most frequent words in a language should require the least effort is that even within words of the same length, the most frequent ones should be easiest to produce and understand. This suggests that more frequent words should have lower phonotactic complexity. After controlling for word length, this is exactly what Mahowald et al. (2018) found in a study of 96 languages, using orthographic probabilities from Wikipedia as a proxy for phonotactic complexity. However, orthography can differ significantly from pronunciation—this correlation has not been confirmed with phonotactic probabilities from phonetic transcriptions. Following Mahowald et al. (2018), we will control for word length as a potential mediator in the relationship between phonotactic complexity and word frequency.

2.5 Morphological Irregularity vs. Length

We are not aware of previous work on the relationship between morphological irregularity and word length, although it is intuitively plausible that one influences the other. For example, a negative correlation within a language could arise because regular inflectional morphology involves combining multiple morphemes, causing words with regular morphology to be longer. Previous work also implies that frequency has an effect on both morphological irregularity and word length. Therefore, we control for frequency as a common cause in assessing the potential relationship between morphological irregularity and word length.

2.6 Length vs. Frequency

Zipf (1935) observed that the most frequent words in a language tend to be short. Since then, the inverse relationship between word length and frequency has been studied in depth and found to follow Zipf's law extremely systematically (see Piantadosi, 2014 for a review), although it is unclear whether word length correlates more strongly with surprisal (Piantadosi et al., 2011) or frequency (Meylan and Griffiths, 2021).

3 Methods

3.1 Data

Our morphological data comes from the UniMorph project, a database of morphologically annotated corpora for 182 languages (Batsuren et al., 2022). Each inflected form is annotated with its lemma

(the lexical meaning) and a set of morphological features; *walked* would be annotated as [VERB; SINGULAR; PAST], for example. While UniMorph provides data for a large set of languages in a universal schema, it does not provide phonetic transcriptions. Because we are interested in how morphology interacts with phonotactics, we use grapheme to phoneme models from Epitran (Mortensen et al., 2018) to convert orthographic transcriptions to IPA. Languages with no available Epitran model were excluded from our analyses.

For training models of phonotactic complexity, we use NorthEuraLex (Dellert et al., 2020), a database of phonetic transcriptions of 1,016 basic concepts for 107 Northern Eurasian languages also studied by Pimentel et al. (2020). For languages that are not included in NorthEuraLex, we use WikiPron (Lee et al., 2020), a database of pronunciation dictionaries from Wiktionary. Languages not in either NorthEuraLex or WikiPron are excluded.

Frequency data is retrieved from Wikipedia,³ and calculated as log count per million words. Following Wu et al. (2019), we exclude all forms with zero frequency, which can differ by orders of magnitude in their true frequency (Baayen, 2001), and we exclude 15 languages where the average probability of the morphological irregularity model predicting the correct surface form is below 0.75. The UniMorph dataset for each language varies in size from 77 to 50,284,287 forms and 37 to 824,074 lemmas. Many of the excluded languages are those with smaller datasets, where the model does not have enough information to accurately predict surface forms.

The languages included in our analysis are: Albanian, Amharic, Azerbaijani, Catalan, Chewa, Czech, Dutch, English, French, German, Hungarian, Italian, Kazakh, Khalka Mongolian, Polish, Portuguese, Romanian, Russian, Serbo-Croatian, Spanish, Swedish, Turkish, Ukrainian, Uzbek, and Zulu. Further details about datasets used can be found in App. A.

3.2 Morphological Irregularity Models

While a binary distinction between regular and irregular morphology is useful in many theories of grammar, a more fine-grained quantitative measure of irregularity is needed to examine the potential relationships with other variables we are interested in. For example, an English speaker might judge

a verb like *walked* to be more regular than *sang*, which is, in turn, more regular than *went*. Wu et al. (2019) define an information theoretic measure of irregularity that captures such intuitions, and is applicable across languages.

This measure, called *degree of morphological irregularity*, abbreviated as MI, is defined using a probabilistic model. Let Σ be an alphabet of symbols⁴ and let \mathcal{S} be a finite set of morphological features, e.g., those provided by the UniMorph dataset. Furthermore, we define the inflector function $\iota: \Sigma^* \times \mathcal{S} \rightarrow \Sigma^*$ that maps a pair of a lemma and a set of morphological features to an inflected surface form, i.e., $(\ell, \sigma) \mapsto w$. The inflector ι is assumed to only operate on *known* lemmas. Thus, to get at a notion of morphological irregularity, we also require a probabilistic inflection model $p(w \mid \ell, \sigma, \mathcal{L}_{-\ell})$, a probability distribution over Σ^* conditioned on a lemma $\ell \in \Sigma^*$, a slot $\sigma \in \mathcal{S}$, and a lexicon with the target lemma removed $\mathcal{L}_{-\ell}$, that tells us which forms in Σ^* are probable inflected surface forms for the lemma ℓ with morphological features σ . The distribution $p(w \mid \ell, \sigma, \mathcal{L}_{-\ell})$ essentially corresponds to a wug-test probability (Berko, 1958), i.e., it tells us the likelihood of the model predicting the correct inflected form of a word it has never seen. To make the probabilities given by $p(w \mid \ell, \sigma, \mathcal{L}_{-\ell})$ more interpretable, Wu et al. (2019) use the negative log odds of probability of the correct surface form, i.e.,

$$\text{MI}(w, \ell, \sigma) = -\log \frac{p(w \mid \ell, \sigma, \mathcal{L}_{-\ell})}{1 - p(w \mid \ell, \sigma, \mathcal{L}_{-\ell})} \quad (1)$$

We can interpret $\text{MI}(w, \ell, \sigma)$ as follows. We achieve a $\text{MI}(w, \ell, \sigma)$ of 0 if the probability of the correct surface form is exactly 0.5, a negative $\text{MI}(w, \ell, \sigma)$ when a surface form is more predictable, and a positive $\text{MI}(w, \ell, \sigma)$ when the form is less predictable.

Morphological irregularity can be considered either a property of an individual word, as in Eq. (1), or as a property of an entire lemma. We calculate the MI of a lemma as the mean MI score of all words in the lemma, i.e.,

$$\text{MI}(\ell) = \frac{1}{|\mathcal{S}|} \sum_{\sigma \in \mathcal{S}} \text{MI}(\iota(\ell, \sigma), \ell, \sigma) \quad (2)$$

Estimating $p(w \mid \ell, \sigma, \mathcal{L}_{-\ell})$ from data. Because the probability distribution $p(w \mid \ell, \sigma, \mathcal{L}_{-\ell})$ is con-

³Retrieved 4/16/23 from <https://dumps.wikimedia.org>.

⁴The symbols could be graphemes or phonemes, depending on the nature of the data annotation.

ditioned on a lexicon with the target lemma removed, the most accurate estimate of MI would require training a separate model for each target lemma in each language. In practice, MI is estimated by training models on a language with a set of lemmas removed, rather than just one. We train neural network models on the UniMorph data using code from Wu et al. (2019), which implements a monotonic hard attention string-to-string induction model described by Wu and Cotterell (2019). In our experiments, we use the same model architecture and training parameters as Wu et al. (2019), and split the lemmas for each language into thirty sets.

3.3 Phonotactic Complexity Models

Similar to our estimation of morphological irregularity, to estimate phonotactic complexity, we take a probabilistic approach. Following Pimentel et al. (2020), we consider a probability distribution $p(w | \mathcal{L})$ over Σ^* . Then, given a word $w \in \Sigma^*$, we define the degree of phonotactic complexity, abbreviated as PC, as follows

$$\text{PC}(w) = -\frac{\log p(w | \mathcal{L})}{|w|} \quad (3)$$

where $|w|$ is the length of the word w . Like the degree of morphological irregularity, PC is a surprisal-based metric that lends itself to easy interpretation. Specifically, if $\text{PC}(w)$ is lower, it means that w is less surprising and therefore more regular.

Estimating $p(w | \mathcal{L})$ from data. The distribution $p(w | \mathcal{L})$ is a hypothetical construct that tells us the probability of an *unknown* word. When we estimate $p(w | \mathcal{L})$ from data, we cannot use the estimated distribution to judge the complexity of those words in the training data. We split training data for each language into ten sets and train ten models, each with one set held out. PC is evaluated on the held-out set for each model. We use the model architecture and training procedure used in code provided by Pimentel et al. (2020), which implements a character-level LSTM (Hochreiter and Schmidhuber, 1997) language model with each phoneme represented by a set of phonetic features from Phoible (Moran et al., 2014).

3.4 Analysis

Following previous studies on the interactions between phonotactic complexity, morphological irregularity, word length, and frequency, we report regression coefficients for each language and each

Y	X	Controls	Rand. Effs.
MI	PC	<u>FR</u> + mean(PC) + <u>WL</u> + mean(WL)	PC + FR + WL
PC	WL	mean(WL)	WL
MI	FR	–	FR
PC	FR	<u>WL</u> + mean(WL)	FR + WL
MI	WL	<u>FR</u> + mean(WL)	WL + FR
WL	FR	–	FR

Table 1: Controls and random effects included in regression models of properties X and Y of words (one row per word): MI = morphological irregularity; PC = phonotactic complexity; WL = word length; FR = frequency. Mean(X) is a language’s average value of X , across all its words. In lme4 syntax, models using words from all languages are: $Y \sim X + \text{Controls} + (1 + \text{Random Effects} | \text{language})$, and individual language models are: $Y \sim X + \text{UnderlinedControls}$.

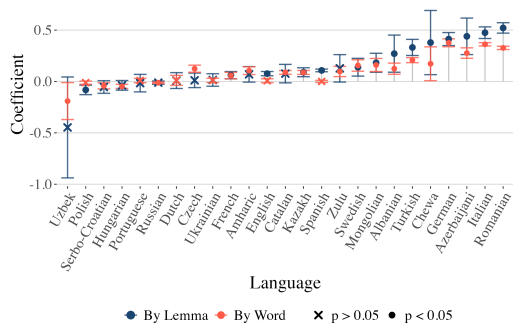
pair of variables, controlling for any necessary variables as described in Tab. 1—these are analogous to partial correlations between the variables of interest. The p -values have been adjusted for multiple comparisons using the Benjamini-Hochberg method. We also perform a linear mixed effects regression analysis across languages for each pair of variables, including random intercepts and slopes for the effect of language. The means of the dependent variable and controls within each language are also included as an additional predictor, to separate across-language effects from within-language effects (see Sonderegger, 2023, §8.10.2.3 and Antonakis et al., 2021). Because languages should not differ in mean word frequency, this predictor is excluded. All predictors were standardized, and log counts-per-million were used for frequencies.

For morphological irregularity analyses (where $Y = \text{MI}$ in Tab. 1), we also report a regression model with data grouped by lemma, where phonotactic complexity and word length are taken as the average within each lemma, frequency is the sum of frequencies within each lemma, and morphological irregularity is calculated according to Eq. (2). Results for the effects of interest in these regression analyses are reported below. Plots of regression predictions and raw data for select languages are shown in App. B.

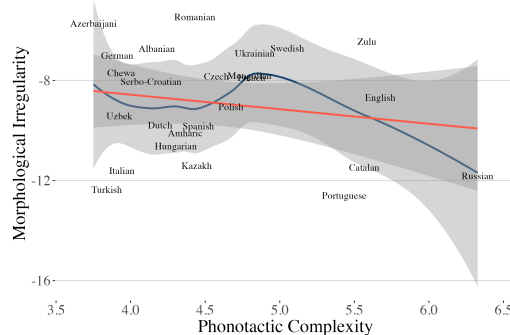
4 Results

4.1 Phonotactic Complexity and Morphological Irregularity

Within languages, we find a positive effect of phonotactic complexity on morphological irreg-

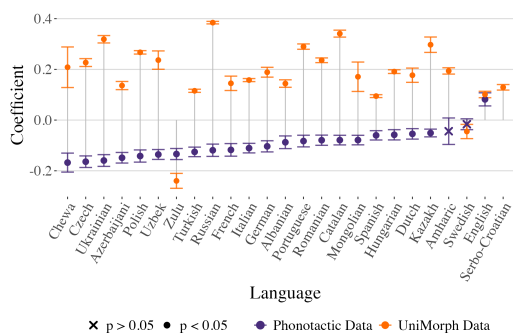


(a) Regression coefficients by language, grouped by lemma and by word, with 95% CIs.

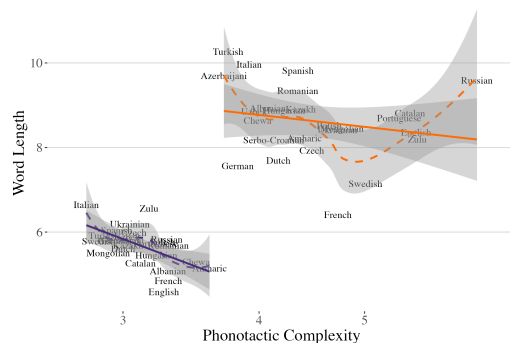


(b) By-language means, with linear (red/light) and LOESS (blue/dark) smoothers.

Figure 1: Phonotactic complexity and morphological irregularity.



(a) Regression coefficients by language, for phonotactic and UniMorph data, with 95% CIs.



(b) By-language means for phonotactic data (purple/dark) and UniMorph data (orange/light), with linear (solid) and LOESS (dotted) smoothers.

Figure 2: Phonotactic complexity and word length.

ularity after controlling for word length and frequency of 0.10 (95% CI [0.05, 0.15], $p < 0.001$, $\sigma = 0.188$)⁵ and a non-significant ($p > 0.05$) effect of mean phonotactic complexity of -0.19 (95% CI [-0.50, 0.12], $p = 0.224$). When grouped by lemma, there is an estimated effect of 0.14 (95% CI [0.07, 0.21], $p < 0.001$, $\sigma = 0.261$) and a non-significant effect of mean phonotactic complexity of 0.00 (95% CI [-0.36, 0.37], $p = 0.993$). Although a majority of languages have a positive effect, as shown in Fig. 1a, some are negative or non-significant. Across languages, there is no evidence of a relationship between mean morphological irregularity and mean phonotactic complexity—we find a non-significant Spearman’s correlation of -0.045 ($p = 0.832$), shown in Fig. 1b.

4.2 Phonotactic Complexity and Word Length

As shown in Fig. 2a, we find a positive relationship between phonotactic complexity and word length within the majority of languages in the UniMorph

data set. We find a similar prediction from the linear model: the estimated effect is 0.18 (95% CI [0.13, 0.23], $p < 0.001$, $\sigma = 0.123$). Across languages, we find no evidence of a correlation between mean phonotactic complexity and mean word length ($\rho = -0.333$, $p = 0.104$), as shown in Fig. 2b. Similarly, the linear model estimates no significant effect of mean word length ($\beta = -0.40$, 95% CI [-0.95, 0.14], $p = 0.145$). The positive effects found in most languages are opposite to the direction predicted by Pimentel et al. (2020) and Pellegrino et al. (2011). However, it is important to note that nearly all words in UniMorph are morphologically complex, while NorthEuraLex (the dataset used by Pimentel et al. (2020)), contains mostly morphologically simple words. As previously noted, morpheme boundaries can create low-probability phonotactic junctures, resulting in higher phonotactic complexity. This suggests that the relationship between word length and phonotactic complexity may be dependent on morphological complexity. We also note that the UniMorph

⁵We use σ to refer to random effect standard deviation.

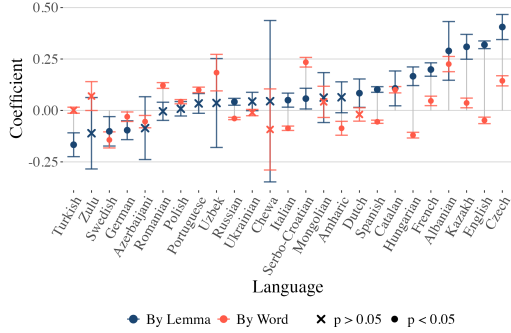


Figure 3: Morphological irregularity and frequency coefficients by language, grouped by lemma and by word, with 95% CIs.

data set contains significantly longer words than NorthEuraLex, suggesting a potential non-linear effect.

To test this, we evaluate the phonotactic complexity of the NorthEuraLex and WikiPron data used to train the phonotactic models and fit models for each language. These results are shown in Fig. 2a, where we can see that most languages show a negative effect in morphologically simple data, replicating the results of Pimentel et al. (2020). We find a Spearman’s correlation coefficient of -0.578 ($p = 0.003$) between average PC and average word length across languages, as shown in Fig. 2b. These results suggest that the finding that phonotactic complexity is negatively correlated with word length only holds for either morphologically simple or relatively short words.

4.3 Morphological Irregularity and Frequency

On the one hand, when morphological irregularity is considered a property of individual words, we find that 10/25 languages have a significant positive effect of frequency on morphological irregularity, and 9/25 have a significant negative effect (Fig. 3). In the linear mixed-effects model, we find no significant effect ($\beta = 0.02$, 95% CI $[-0.02, 0.06]$, $p = 0.224$, $\sigma = 0.100$). On the other hand, when morphological irregularity is considered a property of lemmas rather than individual words, we find that 12/25 languages have a positive effect and only 3/25 have a negative effect. These correlations are shown in Fig. 3. The linear mixed-effects model predicts a positive effect of 0.08 (95% CI $[0.02, 0.13]$, $p = 0.005$, $\sigma = 0.135$), consistent with those of Wu et al. (2019) who examined the same relationship, although using the original or-

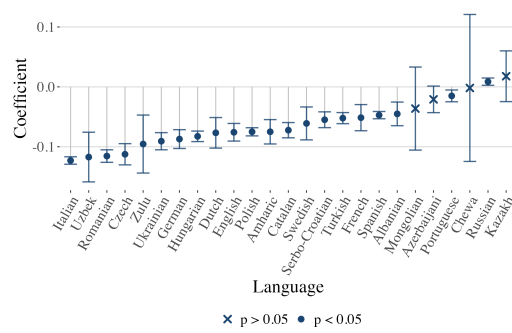


Figure 4: Phonotactic complexity and frequency regression coefficients by language, with 95% CIs.

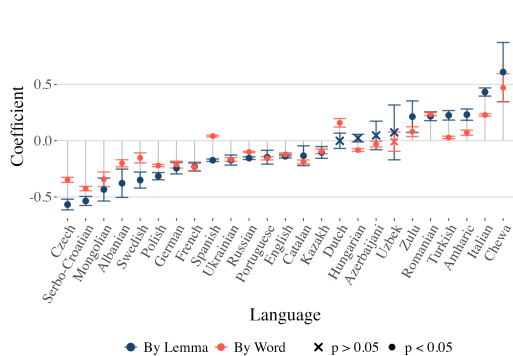
thographic transcriptions from UniMorph, rather than G2P transcriptions. Although the direction of correlation varies across individual languages, there is a tendency towards a positive correlation.

4.4 Phonotactic Complexity and Frequency

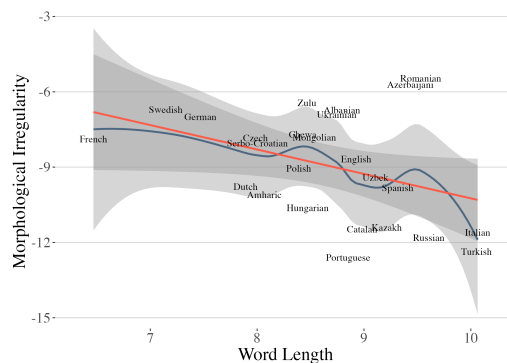
Within most languages, we find a significant negative effect of frequency on phonotactic complexity after controlling for word length, as shown in Fig. 4. The linear model predicts an effect of -0.06 (95% CI $[-0.08, -0.05]$, $p < 0.001$, $\sigma = 0.034$). These findings are similar to those of Mahowald et al. (2018), who found a negative or non-significant correlation in 100% of languages using orthographic data and a simpler measure of phonotactic complexity. Using phonetic transcriptions, we find significantly positive effects in only 1/25 languages, suggesting that this effect is fairly consistent across languages.

4.5 Morphological Irregularity and Length

Within most languages, we find a negative effect of word length on morphological irregularity after controlling for frequency, shown in Fig. 5a. The linear model also estimates a negative effect of -0.07 (95% CI $[-0.15, 0.00]$, $p = 0.058$, $\sigma = 0.193$), although it is non-significant. This effect is also somewhat consistent across languages, as shown in Fig. 5b. Across languages, we find a non-significant Spearman’s correlation of -0.38 ($p = 0.061$), while the linear model estimates the effect of mean word length to be -0.60 (95% CI $[-1.09, -0.12]$, $p = 0.015$). When grouped by lemma, we find a non-significant effect of -0.08 (95% CI $[-0.19, 0.03]$, $p = 0.131$, $\sigma = 0.277$), and a non-significant effect of mean word length of -0.59 (95% CI $[-1.28, 0.11]$, $p = 0.098$).



(a) Regression coefficients by language, grouped by lemma and by word, with 95% CIs.



(b) By-language means, with linear (red/light) and LOESS (blue/dark) smoothers.

Figure 5: Morphological irregularity and word length.

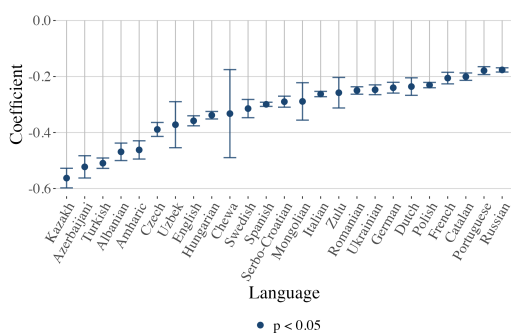


Figure 6: Word length and frequency regression coefficients by language, with 95% CIs.

4.6 Word Length and Frequency

Finally, our results for word length and frequency are exactly as expected, following the prediction of Zipf (1935). We find consistently negative effects of frequency on word length, shown in Fig. 6, and an effect of -0.32 (95% CI $[-0.36, -0.27]$, $p < 0.001$, $\sigma = 0.109$) in the linear model.

5 Discussion

In examining the interactions between phonotactic complexity, morphological irregularity, word length, and frequency, we have found several unexpected and unintuitive results.

Within-language Results. Within most languages, we find significant effects for every pair of variables, although the direction of the effect varies. In our analysis of phonotactic complexity and frequency, we see a strong tendency towards negative effects. In our analysis of morphological irregularity and word length, we see a tendency towards a negative effect, but a positive effect in several languages. In analyses of morphological irregularity and frequency, and of phonotactic complexity

and morphological irregularity, we see a strong tendency towards positive effects. However, the only analysis that is without exception in all languages examined is word length and frequency (by far the best supported by previous work, i.e., Zipf’s Law), where there is a negative effect for all languages. For phonotactic complexity and word length, we found that the direction of the effect for each language changes with the data used. These results complicate the claims of previous work examining several of these pairs across languages, which generally conclude that there is strong support for the relationship. Our results for phonotactic complexity and morphological irregularity also contradict those of Hay and Baayen (2003), Hay (2003), and Burzio (2002). We find evidence of a positive effect of phonotactic complexity on morphological irregularity within language, rather than the negative effect that has previously been argued for. Our analysis controlled for word length and frequency, while previous work did not.

Across-language Results. Across languages, however, we find a negative effect of word length on morphological irregularity and a positive effect of word length on phonotactic complexity, although the direction of this effect changes with the data set used. We also find no evidence of a linear effect of phonotactic complexity on morphological irregularity across languages. However, as can be seen in Fig. 1b, there may be a nonlinear effect across languages. In exploratory data analysis, we also identified several possible nonlinear effects within languages. We leave fully describing any such relationships to future work.

Incomplete Picture. The results presented here suggest that, although these four variables do influence each other, we do not have enough infor-

mation to make claims about one compensating for another, either within a single language or universally across languages. The effects found are potentially consistent with several causal models. It is the causal effects captured by these models that we are interested in—how much does an intervention in one variable *directly* affect another, if at all? If a decrease in morphological irregularity causes a decrease in word length, which then causes an increase in phonotactic complexity, there is no compensatory relationship between morphological irregularity and phonotactic complexity, no matter how correlated they appear to be.

Causal Modeling. Our results assume that the underlying causal structure implied by previous work is correct—that morphological irregularity and phonotactic complexity have a common cause of frequency and word length, that the effect of frequency on phonotactic complexity is mediated by word length, and that morphological irregularity and word length have a common cause of frequency. However, it is possible that a different causal model underlies this data. We leave proposing such causal models and testing their implications for future work. While the models discussed in this work provide a starting point for understanding the structure of the lexicon, evidence supporting the underlying causal structure responsible for generating the data is necessary to evaluate any compensatory relationships in a set of highly correlated variables (Pearl et al., 2016).

Acknowledgements

We thank the Montreal Computational & Quantitative Linguistics Lab for helpful feedback. The first author was supported by funding from the Fonds de recherche du Québec - Société et culture (FRQSC). The third author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (RGPIN-2023-04873). The senior author also gratefully acknowledges the support of the Natural Sciences and Engineering Research Council of Canada and the Canada CIFAR AI Chairs Program. This research was enabled in part by support provided by Calcul Québec and the Digital Research Alliance of Canada.

References

John Antonakis, Nicolas Bastardoz, and Mikko Rönkkö. 2021. *On ignoring the random effects assumption*

in multilevel models: Review, critique, and recommendations. Organizational Research Methods, 24(2):443–483.

R. Harald Baayen. 2001. *Word Frequency Distributions*, volume 18 of *Text, Speech and Language Technology*. Springer Dordrecht.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, Jaime Rafael Montoya Samame, Delio Siticonatzi Camaiteri, Esaú Zumaeta Rojas, Didier López Francis, Arturo Oncevay, Juan López Bautista, Gema Celeste Silva Villegas, Lucas Torroba Hennigen, Adam Ek, David Guriel, Peter Dirix, Jean-Philippe Bernardy, Andrey Scherbakov, Aziyana Bayyr-ool, Antonios Anastasopoulos, Roberto Zariquiey, Karina Sheifer, Sofya Ganieva, Hilaria Cruz, Ritván Karahóga, Stella Markantonatou, George Pavlidis, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Candy Angulo, Jatayu Baxi, Andrew Krizhanovsky, Natalia Krizhanovskaya, Elizabeth Salesky, Clara Vania, Sardana Ivanova, Jennifer White, Rowan Hall Maudslay, Josef Valvoda, Ran Zmigrod, Paula Czarnowska, Irene Nikkarinen, Aelita Salchak, Brijesh Bhatt, Christopher Straughn, Zoey Liu, Jonathan North Washington, Yuval Pinter, Duygu Ataman, Marcin Wolinski, Totok Suhardijanto, Anna Yablonskaya, Niklas Stoehr, Hossep Dolatian, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Aryaman Arora, Richard J. Hatcher, Ritesh Kumar, Jeremiah Young, Daria Rodionova, Anastasia Yemelina, Taras Andrushko, Igor Marchenko, Polina Mashkovtseva, Alexandra Serova, Emily Prud'hommeaux, Maria Nepomniashchaya, Fausto Giunchiglia, Eleanor Chodroff, Mans Hulden, Miikka Silfverberg, Arya D. McCarthy, David Yarowsky, Ryan Cotterell, Reut Tsarfaty, and Ekaterina Vylomova. 2022. *UniMorph 4.0: Universal Morphology*. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855. European Language Resources Association.

Jean Berko. 1958. *The child's learning of English morphology. Word*, 14(2-3):150–177.

Luigi Burzio. 2002. *Missing players: Phonology and the past-tense debate. Lingua*, 112(3):157–199.

Joan L. Bybee. 1985. *Morphology: A Study of the Relation between Meaning and Form*. John Benjamins.

Christophe Coupé, Yoon Mi Oh, Dan Dediu, and François Pellegrino. 2019. *Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. Science Advances*, 5(9):eaaw2594.

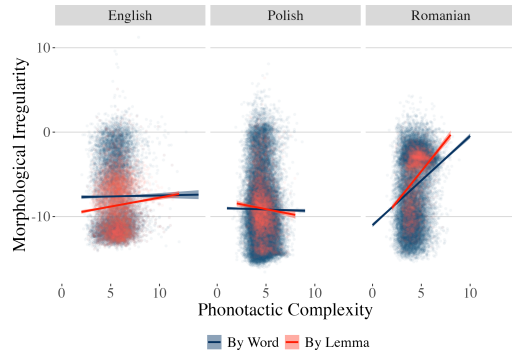
- Johannes Dellert, Thora Daneyko, Alla Münch, Alina Ladygina, Armin Buch, Natalie Clarius, Ilja Grigorjew, Mohamed Balabel, Hizniye Isabella Boga, Zalina Baysarova, Roland Mühlenbernd, Johannes Wahle, and Gerhard Jäger. 2020. [NorthEuraLex: a wide-coverage lexical database of Northern Eurasia](#). *Language Resources and Evaluation*, 54:273–301.
- Viviana Fratini, Joana Acha, and Itziar Laka. 2014. Frequency and morphological irregularity are independent variables. Evidence from a corpus study of Spanish verbs. *Corpus Linguistics and Linguistic Theory*, 10(2):289–314.
- Jennifer Hay. 2003. *Causes and Consequences of Word Structure*. Routledge.
- Jennifer Hay and Harald Baayen. 2003. Phonotactics, parsing and productivity. *Italian Journal of Linguistics*, 15:99–130.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- C.F. Hockett. 1955. *A Manual of Phonology*. Waverly Press.
- Jackson L. Lee, Lucas F. E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. 2020. [Massively multilingual pronunciation modeling with WikiPron](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4223–4228. European Language Resources Association.
- Kyle Mahowald, Isabelle Dautriche, Edward Gibson, and Steven T. Piantadosi. 2018. [Word forms are structured for efficient use](#). *Cognitive Science*, 42(8):3116–3134.
- Gary F. Marcus, Steven Pinker, Michael Ullman, Michelle Hollander, T. John Rosen, Fei Xu, and Harald Clahsen. 1992. [Overregularization in language acquisition](#). *Monographs of the Society for Research in Child Development*, 57(4):i–178.
- André Martinet. 1955. *Économie des changements phonétiques: Traité de phonologie diachronique*. A. Francke.
- Clara Meister, Tiago Pimentel, Patrick Haller, Lena Jäger, Ryan Cotterell, and Roger Levy. 2021. [Revisiting the uniform information density hypothesis](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 963–980. Association for Computational Linguistics.
- Stephan C. Meylan and Thomas L. Griffiths. 2021. [The challenges of large-scale, web-based language datasets: Word length and predictability revisited](#). *Cognitive Science*, 45(6):e12983.
- Steven Moran, Daniel McCloy, and Richard Wright, editors. 2014. *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology.
- David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. [Epitran: Precision G2P for many languages](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*. European Language Resources Association.
- Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal Inference in Statistics: A Primer*. John Wiley & Sons.
- François Pellegrino, Christophe Coupé, and Egidio Marsico. 2011. [A cross-language perspective on speech information rate](#). *Language*, 87(3):539–558.
- Steven T. Piantadosi. 2014. [Zipf’s word frequency law in natural language: A critical review and future directions](#). *Psychonomic Bulletin & Review*, 21(5):1112–1130.
- Steven T. Piantadosi, Harry Tily, and Edward Gibson. 2011. [Word lengths are optimized for efficient communication](#). *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.
- Tiago Pimentel, Clara Meister, Ethan Wilcox, Kyle Mahowald, and Ryan Cotterell. 2023. [Revisiting the optimality of word lengths](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2240–2255. Association for Computational Linguistics.
- Tiago Pimentel, Brian Roark, and Ryan Cotterell. 2020. [Phonotactic complexity and its trade-offs](#). *Transactions of the Association for Computational Linguistics*, 8:1–18.
- Ingo Plag. 2002. [The role of selectional restrictions, phonotactics and parsing in constraining suffix ordering in English](#). In Geert Booij and Jaap Van Marle, editors, *Yearbook of Morphology 2001*, pages 285–314. Springer.
- Morgan Sonderegger. 2023. *Regression Modeling for Linguistic Data*. MIT Press.
- Shijie Wu and Ryan Cotterell. 2019. [Exact hard monotonic attention for character-level transduction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537. Association for Computational Linguistics.
- Shijie Wu, Ryan Cotterell, and Timothy O’Donnell. 2019. [Morphological irregularity correlates with frequency](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5117–5126. Association for Computational Linguistics.
- Charles Yang. 2016. *The Price of Linguistic Productivity: How Children Learn to Break the Rules of Language*. MIT press.
- George Kingsley Zipf. 1935. *The Psycho-Biology of Language: An Introduction to Dynamic Philology*. Houghton Mifflin.

A Languages in Analysis

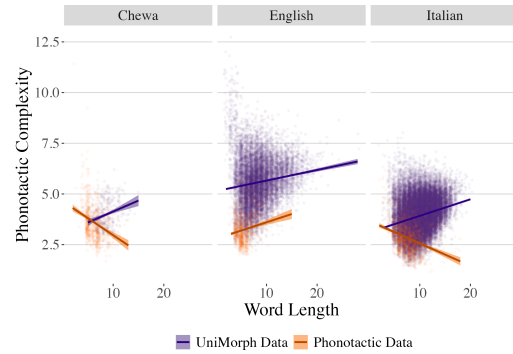
Language	Family	Type	Forms	Lemmas	Phon.	Acc.
Albanian	Indo-Eur.	Fus.	33483	589	NL	81.16
Amharic	Afro-Asiatic	Fus.	46224	2461	WP	92.82
Azerbaijani	Turkic	Agg.	8004	340	NL	76.15
Catalan	Indo-Eur.	Fus.	81576	1547	NL	95.88
Chewa	Atl. Congo	Agg.	4370	227	WP	85.91
Czech	Indo-Eur.	Fus.	50284287	824074	NL	89.89
Dutch	Indo-Eur.	Fus.	55467	4993	NL	92.33
English	Indo-Eur.	Fus.	115523	22765	NL	80.89
French	Indo-Eur.	Fus.	367732	7535	NL	84.96
German	Indo-Eur.	Fus.	179339	15060	NL	86.95
Hungarian	Uralic	Agg.	490394	13989	NL	93.27
Italian	Indo-Eur.	Fus.	509574	10009	NL	98.85
Kazakh	Turkic	Agg.	40283	1755	NL	98.00
Khalka Mongolian	Mongolic	Agg.	30143	2140	NL	82.35
Polish	Indo-Eur.	Fus.	13882543	274550	NL	90.93
Portuguese	Indo-Eur.	Fus.	303996	4001	NL	97.59
Romanian	Indo-Eur.	Fus.	80266	4405	NL	78.09
Russian	Indo-Eur.	Fus.	473481	28068	NL	95.53
Serbo-Croatian	Indo-Eur.	Fus.	840799	24419	NL	92.30
Spanish	Indo-Eur.	Fus.	382955	5460	NL	83.35
Swedish	Indo-Eur.	Fus.	78411	10553	NL	83.23
Turkish	Turkic	Agg.	275460	3579	NL	96.07
Ukrainian	Indo-Eur.	Fus.	20904	1493	NL	85.11
Uzbek	Turkic	Agg.	810	68	WP	87.86
Zulu	Atl. Congo	Agg.	49562	621	WP	75.45
<i>Bengali</i>	Indo-Eur.	Fus.	4443	136	NL	37.75
<i>Cebuano</i>	Austronesian	Agg.	618	97	WP	54.17
<i>Hindi</i>	Indo-Eur.	Fus.	54438	258	NL	68.79
<i>Indonesian</i>	Austronesian	Agg.	27714	3877	WP	49.26
<i>Kabardian</i>	Abkhaz-Adyge	Agg.	3092	250	WP	63.36
<i>Kashubian</i>	Indo-Eur.	Fus.	509	37	WP	1.43
<i>Kyrgyz</i>	Turkic	Agg.	5544	98	WP	29.81
<i>Maltese</i>	Afro-Asiatic	Fus.	3584	112	WP	29.82
<i>Swahili</i>	Atl. Congo	Fus.	14130	185	WP	35.17
<i>Tagalog</i>	Austronesian	Agg.	2912	344	WP	46.22
<i>Tajik</i>	Indo-Eur.	Fus.	77	75	WP	27.01
<i>Telugu</i>	Dravidian	Agg.	1548	127	NL	41.67
<i>Turkmen</i>	Turkic	Agg.	810	68	WP	30.42
<i>Urdu</i>	Indo-Eur.	Fus.	12572	182	WP	50.89
<i>Uyghur</i>	Turkic	Agg.	8178	90	WP	15.12

Table 2: All languages in analysis. *Italicized* languages are excluded. Abbreviations: Fus. - Fusional; Agg. - Agglutinative; Phon. - Phonotactic data source; NL - NorthEuraLex; WP - WikiPron; Acc. - Morphology Model Accuracy

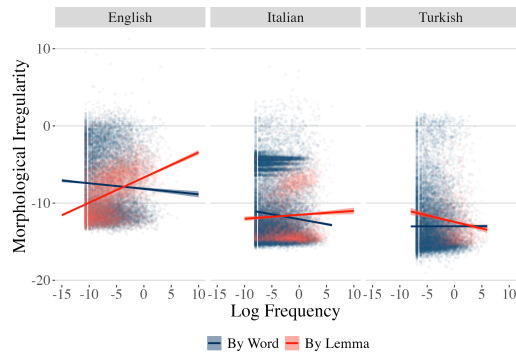
B Plots of Individual Languages



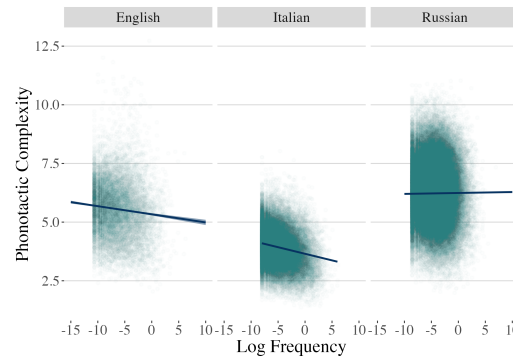
(a) Morphological irregularity and phonotactic complexity data for English, Polish, and Romanian, with linear model predictions from Tab. 1 and 95% CIs.



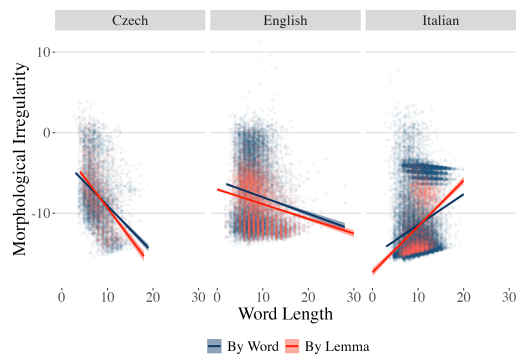
(b) Phonotactic complexity and word length data for Chewa, English, and Italian, with linear model predictions from Tab. 1 and 95% CIs.



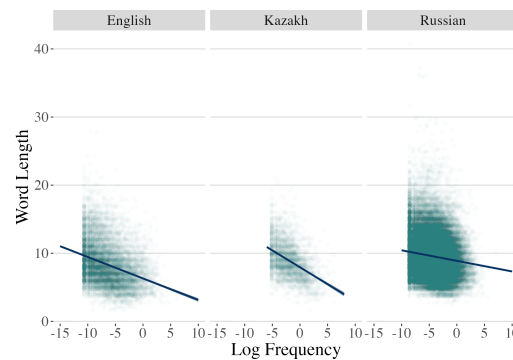
(c) Morphological irregularity and frequency data for English, Italian, and Turkish, with linear model predictions from Tab. 1 and 95% CIs.



(d) Phonotactic complexity and frequency data for English, Italian, and Russian, with linear model predictions from Tab. 1 and 95% CIs.



(e) Morphological irregularity and word length data for Czech, English, and Italian, with linear model predictions from Tab. 1 and 95% CIs.



(f) Word Length and frequency data for English, Kazakh, and Russian, with linear model predictions from Tab. 1 and 95% CIs.

Algebraic Reanalysis of Phonological Processes Described as Output-Oriented

Dakotah Lambert

Université Jean Monnet Saint-Étienne, CNRS
 Institut d'Optique Graduate School
 Laboratoire Hubert Curien
 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

Many phonological processes – including exemplars of local harmony, iterative spreading, and long-distance harmony patterns – have been shown to belong to the Output (Tier-based) Strictly Local (O(T)SL) functions. This article provides an algebraic analysis of these processes. The algebraic approach to subregular pattern complexity is important because it unifies the computational characterizations of constraints and processes, while leveraging a wealth of results in theoretical computer science on the structural properties of these classes. These structural properties are useful because they underlie algorithms for classifying and learning.

The first result shows that the O(T)SL class has no corresponding algebraic characterization. The second result establishes that canonical examples of these processes belong to the definite and reverse definite classes, or their tier-based extensions, some of the simplest algebraic classes. The third result provides a single learning algorithm for these classes which identifies them in the limit from positive data.

1 Introduction

Local harmony, iterative spreading, and long-distance harmony processes are ubiquitous in natural languages and have been the subject of much linguistic (Walker, 1998, 2011, 2014; Rose and Walker, 2004; Hansson, 2010; Nevins, 2010; van der Hulst, 2018) and computational (Heinz et al., 2011; Heinz and Lai, 2013; Chandlee et al., 2015; Aksënova and Deshmukh, 2018; Burness and McMullin, 2019; Burness et al., 2021; Lambert, 2023) research. Much of this latter work studies the computational properties of these processes when viewed as string-to-string functions.

Following Filiot et al. (2019), we consider algebraic analyses of such functions. Each function can be associated with a semigroup and classified



Figure 1: Two analyses of iterative spreading

according to properties of that semigroup. For example, Lambert and Heinz (2023) proved that, considering only total functions, the input strictly local functions (ISL) (Chandlee et al., 2014) are precisely the algebraic variety of **definite** functions.

Output Strictly Local (OSL) functions (Chandlee et al., 2015) are one way to characterize and represent phonological processes such as local harmony and iterative spreading (Chandlee and Heinz, 2018), and when combined with tiers, long-distance harmony (Burness et al., 2021). These processes are commonly understood as **output-oriented** because the output at any given point appears to depend on some prior output.

As an example, consider iterative spreading in Johore Malay where /pəŋawasan/ ‘supervision’ is pronounced as [pəŋãwãsan] with nasalization on a successive sequence of vowels and semivowels (Heinz, 2010). Consider the rules shown below.

$$[-\text{cons}] \rightarrow [+nas]/[+nas]_ \quad (1)$$

$$[-\text{cons}] \rightarrow [+nas]/[+nas] [-\text{cons}]^* _ \quad (2)$$

In order for Rule 1 to account for the nasal iterative spreading in Malay, it must apply iteratively from left to right. Consequently, the second [ã] is nasalized because the preceding glide has been nasalized in an earlier iteration of the rule. On the other hand, Rule 2 can apply simultaneously. The analysis with Rule 1 is considered output-oriented, but not the analysis with Rule 2. The applications of Rules 1 and 2 are schematized in Figure 1 in red and blue respectively. This issue is relevant today: Walker (2014) argues on empirical grounds that some harmony processes in some languages should be analyzed in the way suggested by Rule 2,

though her analysis uses Optimality Theory.

This article contains three main results. The first is that every finite semigroup is the syntactic semigroup of some OSL function. Since there are sequential functions that lie outside of this class, no algebraic variety can contain all and only the O(T)SL functions.

The second result is an algebraic analysis of a sample of canonical local harmony, iterative spreading, and long-distance harmony processes. By “canonical”, we mean specific patterns in the literature that served to motivate the Output Strictly Local class and other related classes (Chandlee et al., 2015; Burness et al., 2021). The focus is less on the processes themselves, and more on the algebraic techniques by which they are classified. We make no claim that these processes are representative of the most complex of attested phenomena, which is often the subject of debate (see for example Kula and Syed, 2020).

For each process, we provide an algebraic analysis and discuss the classes in which it lies. We find that the patterns we consider have their behaviors fixed by either the k most recent symbols encountered or the first k symbols encountered, potentially projected onto a tier. This corresponds to the (tier-based) definite or (tier-based) reverse-definite classes of formal languages. In other words, the actual processes the O(T)SL functions were introduced to describe actually belong to some of the simplest and most restrictive algebraic classes. Furthermore, in the case of iterative spreading and long-distance harmony, the algebraic analysis indicates an interpretation akin to Rule 2. In this way, this paper provides a deeper insight into processes that have been described as output-oriented in the phonological literature.

Third, we present a learning algorithm, based on the smallest algebraically natural class which includes these functions, and prove it is learnable in the limit from positive data. As such, this algorithm does not take into account the output-oriented nature of the processes considered.

Section 2 recalls some relevant definitions. Then §3 shows how to conduct an algebraic analysis using post-nasal voicing as a running example. Then §4 demonstrates that no algebraic property can distinguish the output (tier-based) strictly local functions from arbitrary other sequential functions. §5 follows by providing algebraic analyses for several other processes that have been analyzed as output-oriented. §6 presents a learning algorithm based

on SOSFIA (Jardine et al., 2014) that is powerful enough to handle the processes considered. Discussion and concluding remarks follow in §7.

2 Preliminaries

This section recalls basic definitions and notation. Given a finite alphabet Σ , let Σ^* denote the set of finite strings over Σ . Let λ denote the string of length 0 and $|w|$ the length of string w . For all strings $w \in \Sigma^*$, define $\text{Suff}_k(w)$ to be the string v if there exists $u \in \Sigma^*$ such that $w = uv$ and $|v| = k$ and to be w otherwise.

A **tier** T is a subset of Σ and the **tier projection** of a string w is defined recursively as follows. For the base case, $\pi_T(\lambda) = \lambda$, and for the inductive case, $\pi_T(wa) = \pi_T(w)a$ iff $a \in T$ and $\pi_T(w)$ otherwise. Symbols in $\Sigma - T$ are called **neutral letters** and symbols in T are called **salient**.

A **semigroup** is a set S closed under an associative multiplication operation. An element a of S is **idempotent** whenever $aa = a$. If all elements of S are idempotent, then S is a **band**.

A **finite-state transducer** is an abstract machine that reads an input sequence, one symbol at a time, and produces one or more sequences as output (Raney, 1958). In this work, we are concerned only with **total, sequential** transducers, the subset of these machines in which computation is deterministic and each input sequence produces one and only one output sequence (Schützenberger, 1977). Formally, such a machine is a 8-tuple: $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, \rho, \sigma \rangle$, where Q is a finite set of states, Σ a finite set of input symbols, Γ a finite set of output symbols, $\delta: Q \times \Sigma \rightarrow Q \times \Gamma^*$ a transition function, $q_0 \in Q$ an initial state, $\rho \in \Gamma^*$ a prefix prepended to all output sequences, and finally $\sigma: Q \rightarrow \Gamma^*$ a suffixing function.

The machine processing function μ is defined recursively. For the base case, let $\mu(q, \lambda, v) = v\sigma(q)$. The recurrence is given in Equation 3 below where $a \in \Sigma$, $\delta(q, a) = (q', w)$.

$$\mu(q, au, v) = \mu(q', u, vw) \quad (3)$$

Then the function $f: \Sigma^* \rightarrow \Gamma^*$ that \mathcal{A} computes is $f(w) = \mu(q_0, w, \rho)$.

For every sequential function f , there is a unique (up to isomorphism) sequential transducer representing it, which is its **minimal onward form**. Informally, onwardness means the output is produced as early as possible. Readers are referred to Chofrut (2003) for technical details. The transducers

introduced in this article, upon which the algebraic analyses are based, are all in minimal onward form.

Sequential functions come in two types: **left-to-right-sequential** and **right-to-left-sequential**. Left-to-right-sequential are defined as above. Right-to-left-sequential functions can be represented by transducers which process the input string from right to left.¹ In general, reversing the direction of the transducer computes the reversal of the process. For example, reversing the direction of a transducer for post-nasal voicing yields pre-nasal voicing, and reversing the direction of a transducer for regressive symmetric harmony yields progressive symmetric harmony.

It will be convenient to refer to the first component of the transition function. Whenever $\delta(q, a) = (q', w)$, we write $q * a = q'$.

For any tier $T \subseteq \Sigma$, a is a neutral letter if and only if for all $q \in Q$ it is the case that $q * a = q$ (Lambert, 2023). In other words, neutral letters are exactly those which never cause state to change.

A function f is **Output Strictly k -Local (k -OSL)** if there is sequential transducer representing f with the property that the current state is entirely determined by the $k - 1$ most recent symbols of output (Chandlee et al., 2015). In terms of the recurrence relation (Equation 3), this means that $q' = \text{Suff}_{k-1}(vw)$.

The **Output Tier-based Strictly k -Local (k -OTSL)** functions are defined in the same way, except that the suffix is taken after projection to a fixed set of salient symbols (Burness et al., 2021). As with sequential functions, O(T)SL functions come in left-to-right and right-to-left variants.

Input Strictly k -Local (k -ISL) functions are also defined similarly where the suffix is taken over the input symbols (Chandlee et al., 2014).²

3 Algebraic Analysis

The algebraic theory of formal languages and functions provides a window into the kind of information to which a perceiver must attend when learning a pattern or when classifying it (Rogers et al., 2012; Filiot et al., 2016, 2019; Lambert, 2022).³ This section explains the fundamentals of algebraic analysis of string-to-string functions using the phonological

¹One way for \mathcal{A} to process w right-to-left is to give \mathcal{A} the reverse of w and then reverse its output.

²The left-to-right and right-to-left ISL functions are the same class.

³A link to open source software for classifying and learning patterns will be provided upon acceptance.

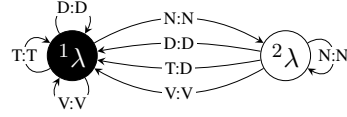


Figure 2: A minimal transducer for post-nasal voicing.

process of post-nasal voicing as a running example.

As an example, consider the phonological process of post-nasal voicing (PNV) in the Puyu Pungo dialect of Quechua, investigated by Burness et al. (2021). Here, a voiceless obstruent directly following a nasal becomes voiced. A transducer in minimal onward form for PNV is shown in Figure 2, where ‘V’ represents a vowel, ‘N’ a nasal consonant, ‘T’ a voiceless obstruent, and ‘D’ a voiced obstruent. States are labeled by an integer index and the output of the suffixing function σ . Edges are labeled with their input and output, in order, separated by a colon. The initial state is black.

Because this transducer is small, visual inspection is sufficient to establish that PNV is both ISL and OSL. The set of length-one input suffixes that lead to state 1 are $\{\$, D, T, V\}$, where $\$$ represents the beginning of the string, while the set of those that lead to state 2 is $\{N\}$. The sets are disjoint; thus, the function is ISL. Exactly the same analysis applies to output suffixes, so the function is OSL.

3.1 Transition Semigroups

Given a finite-state machine, its **transition semigroup** is built from the **actions** of each letter, which are the changes they make to the state space. Formally, given a listing of the states in Q , $\langle q_0, \dots, q_n \rangle$, the action given by $a \in \Sigma$ is the tuple $\langle q_0 * a, \dots, q_n * a \rangle$. Note that distinct symbols may have the same action, which means they exhibit the same behaviors. Importantly, since neutral letters do not change state, their action is always the identity action $\langle q_0, \dots, q_n \rangle$, denoted $\mathbb{1}$.

The actions given by the letters form the **basis** of the transition semigroup. The rest of the semigroup is generated as follows. Given two actions a and b , one constructs the product ab by first applying a , then applying b to its result: $b \circ a$. The product is potentially a new action. However, as there are finitely many states, there are ultimately at most finitely many actions over these states. The transition semigroup is the set of actions generated under this composition, including the basis.

In the transducer for PNV in Figure 2, the list-

Figure 3: Multiplication table (left) and eggbox diagram (right) for post-nasal voicing.

	x	y
x	x	y
y	x	y

x	y
-----	-----

ing of states is given by their index. Observe that $\langle 1 * N, 2 * N \rangle = \langle 2, 2 \rangle$, and $\langle 1 * D, 2 * D \rangle = \langle 1, 1 \rangle$. In fact, there are two actions which arise from individual letters in this transducer: $x = \langle 1, 1 \rangle$ from ‘D’, ‘T’ and ‘V’, and $y = \langle 2, 2 \rangle$ from ‘N’. Thus, x, y are the basis of the transition semigroup.

Recall that successive actions in the transducer translate to multiplication in the semigroup. Observe that $xy = y$ because (i) $x = \langle 1, 1 \rangle$, which means that it maps state 1 to state 1 and state 2 to state 1, (ii) $y = \langle 2, 2 \rangle$ means that it maps state 1 to state 2 and state 2 to state 2, and (iii) when action x is followed by action y , the product action maps state 1 to state 2 and state 2 to state 2, which is the same action as y . Similar reasoning reveals that $xx = x, xy = y, yx = x, \text{ and } yy = y$. Additional multiplication yields no new actions, so this pair of elements makes up the entire syntactic semigroup, shown in Figure 3 (left), where the cell at row x and column y is the product xy . The eggbox diagram shown at right in Figure 3 is another representation of the structure, which will be discussed in more detail in §3.4.

The transition semigroup of a transducer \mathcal{A} in minimal onward form for a sequential function f is the **syntactic semigroup** of \mathcal{A} (Filiot et al., 2016). Since the states of the minimal automaton correspond to minimally distinct behaviors, the syntactic semigroup indicates how input sequencing influences the behavior of f .

Since the transducer in Figure 2 for PNV is in minimal onward form, its transition semigroup is that function’s syntactic semigroup.

3.2 Varieties

A **variety** is a class of semigroups closed under finitary direct products (tuples which multiply pointwise), quotients (structured merges of elements), and inverse nonerasing homomorphisms. Interested readers are referred to Almeida (1995) for more information on these operations in addition to the varieties discussed in this article and others. Pin (1984) discusses the relationship between varieties of semigroups and varieties of formal languages, which can be extended to string-to-string functions

(Lambert, 2022). As a consequence of Eilenberg’s variety theorem (Eilenberg, 1976), many important classes of formal languages and string-to-string functions are characterized by properties of their syntactic semigroup (Pin, 1984; Lambert, 2022). As an example, the class of ISL functions corresponds exactly to the variety of definite semigroups, defined below (Lambert and Heinz, 2023).

3.3 Green’s relations

Many important varieties, including the definite variety, can be expressed in terms of binary relations defined by Green (1951). Given a semigroup S , Colcombet (2011) gives the following preorders.⁴

- $a \leq_{\mathcal{L}} b$ iff $a \in Sb \cup \{b\}$.
- $a \leq_{\mathcal{R}} b$ iff $a \in bS \cup \{b\}$.
- $a \leq_{\mathcal{J}} b$ iff $a \in SbS \cup Sb \cup bS \cup \{b\}$.

Then a is “ \mathcal{L} -related” to b (denoted $a \mathcal{L} b$) if and only if $a \leq_{\mathcal{L}} b$ and $b \leq_{\mathcal{L}} a$. If S contains no pair of distinct elements that are “ \mathcal{L} -related” it is said to be **\mathcal{L} -trivial**. The relations \mathcal{R} and \mathcal{J} , and the properties \mathcal{R} -trivial and \mathcal{J} -trivial, are defined similarly.

A semigroup belongs to the variety **D** of **definite** semigroups if and only if it is \mathcal{L} -trivial and the only idempotent elements lie in the minimal \mathcal{J} -class. Similarly, a semigroup belongs to the variety **K** of **reverse definite** semigroups if and only if it is \mathcal{R} -trivial and the only idempotent elements lie in the minimal \mathcal{J} -class (Almeida, 1995).

Recalling that neutral letters give rise to the identity action $\mathbb{1}$, Lambert (2023) defines a semigroup S to be **tier-based** definite (reverse definite) if and only if the elements of S other than $\mathbb{1}$ satisfy the conditions for definiteness and reverse definiteness. The tier-based definite and reverse definite classes are denoted $\llbracket \mathbf{D} \rrbracket_T$ and $\llbracket \mathbf{K} \rrbracket_T$, indicating the interpretation of the variety on some tier T .

A semigroup’s multiplication table reveals which elements of the semigroup stand in which of Green’s relations. Two elements are \mathcal{R} -related if, in the multiplication table of their semigroup, their rows contain the same set of elements, including the labels (the elements themselves). Figure 3 (left) for PNV shows x and y are \mathcal{R} -related, as each labels a row consisting of the set $\{x, y\}$.

Two elements in a semigroup are \mathcal{L} -related if the columns of the multiplication table contain the

⁴Note $Sb = \{xb : x \in S\}$ and similarly for bS and SbS .

same set of elements, including the labels. In Figure 3 (left), the column of x is $\{x\}$ while that of y is $\{y\}$, so no two distinct elements are \mathcal{L} -related.

Finally, the \mathcal{J} -order is defined such that $x \leq_{\mathcal{J}} y$ if and only if the union of the columns specified in the row of x is a subset of the union of the columns specified in the row of y . In Figure 3 (left), the row for x is $\{x, y\}$, and the union of those columns is $\{x\} \cup \{y\} = \{x, y\}$. The same holds for y , so $x \leq_{\mathcal{J}} y$ and $y \leq_{\mathcal{J}} x$. Thus $x \mathcal{J} y$.

Synthesizing, no two elements in the syntactic semigroup for PNV are \mathcal{L} -related (it is \mathcal{L} -trivial). Also, x and y are idempotents and they are \mathcal{J} -related and thus belong to the same \mathcal{J} -class. This \mathcal{J} -class is minimal since it is the only one. Therefore, this semigroup satisfies the definition of a definite semigroup and belongs to **D**. It does not belong to **K** because two of its elements are \mathcal{R} -related.

This algebraic analysis confirms the earlier ISL analysis. Moreover, it is a band. One consequence is that the degree of definiteness (the suffix length under consideration) is 1 (Lambert, 2022) and therefore the k -value for which it is ISL is 2.

3.4 Eggbox Diagrams

Another useful representation of a semigroup is given by what Clifford and Preston (1961) call the **eggbox diagram**, whose design is based on Green’s relations. The eggbox diagram is constructed as a collection of grids. Within a grid, two elements share a row if and only if they are \mathcal{R} -related. They share a column if and only if they are \mathcal{L} -related. All elements within a grid are equal with respect to the \mathcal{J} -order. Grids are organized into a graph such that an edge exists from one to another if and only if the target is lower with respect to the \mathcal{J} -order than the source. There can be no cycles, so in depictions the source shall always be the higher grid. Finally, idempotent elements have their cells shaded. The eggbox diagram of the syntactic semigroup for PNV is shown in Figure 3 (right). The eggbox diagram makes clear that this semigroup belongs to the definite variety **D** because it shows there is one \mathcal{J} -class, so it is minimal, and its elements are idempotent. Furthermore, every column in this grid is of depth one and so no pair of distinct elements are \mathcal{L} -related. Eggbox diagrams are used for later analyses.

3.5 Directionality

The transducer in Figure 2 operates from left to right. A transducer operating right-to-left which

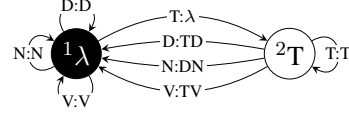


Figure 4: Right-to-left version of post-nasal voicing.

computes the same function does not necessarily have the same structure. Figure 4 depicts a right-to-left machine for the same post-nasal voicing process. It is interesting to observe that this transducer has the same structure as would arise from a left-to-right transducer, in minimal onward form, computing *prenasal* voicing. When a voiceless obstruent is encountered, output is delayed until the following symbol, when it is known whether the output should be voiced or voiceless. The right-to-left computation remains ISL, as the set $\{\$, D, N, V\}$ of suffixes lead to state 1, while the set $\{T\}$ of suffixes lead to state 2. But it does not have an OSL structure, as ‘T’ moves from state 1 to state 2 without outputting anything; the k -suffix of the output is unchanged while the state changes. This demonstrates the well-known fact that output strict locality is directional (Chandlee et al., 2015).

The basis of its syntactic semigroup, however, is the same as before: $x = \langle 1, 1 \rangle$ from ‘D’, ‘N’ and ‘V’, and $y = \langle 2, 2 \rangle$ from ‘T’. Thus, the syntactic semigroup is also the same. This is coincidental and is not generally guaranteed, as witnessed by analyzing iterative spreading in §5.

4 OSL is not Algebraic

Given that algebraic results provide new tools for classifying and learning and that ISL functions correspond exactly to functions with definite semigroups, it is natural to ask what variety, if any, corresponds to OSL functions.

Theorem 1. *For any finite semigroup S , there is an OSL function whose syntactic semigroup is precisely S .*

Proof. Let S be a finite semigroup generated by a basis $B \subseteq S$. Let Γ be an alphabet containing at least two letters. Further, let n be $\lceil \log_{|\Gamma|} |S| \rceil$. Finally let $f: S \rightarrow \Gamma^n$ be an injective function assigning to each element of S a unique arbitrary string in Γ^n . At this point we can construct a sequential transducer $\mathcal{A} = \langle S, B, \Gamma, \delta, 1, \lambda, f \rangle$, where $\delta: S \times B \rightarrow S \times \Gamma^*$ where $\delta(x, y) = \langle xy, f(xy) \rangle$. The output is produced n symbols

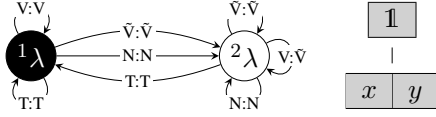


Figure 5: Iterative nasal spreading, with eggbox.

at a time and the state is fixed by the last n output symbols. So this transducer is clearly OSL.

It is also minimal, as every state yields a different output upon termination. Thus the syntactic semigroup of \mathcal{A} is equivalent to its transition semigroup, which by construction is identical to S itself. \square

It follows that for any non-OSL sequential function, there exists an OSL function with the same syntactic semigroup. Thus this class of functions does not correspond to a variety of semigroups and is not well-behaved under Eilenberg’s theory.

5 Analysis of Output-Oriented Processes

We are thus led to ask which algebraic varieties the phonological processes that motivated the OSL class belong to, if any. This section examines canonical attested processes that have been analyzed in an output-oriented way: iterative spreading processes like nasal spreading, and harmony processes, both symmetric and asymmetric, such as sibilant harmony. The algebraic analyses show these processes to be (tier-based) definite or (tier-based) reverse definite. The results of the analysis are summarized in Table 1 (page 7).

5.1 Iterative Spreading

Post-nasal voicing is an example of noniterative assimilation. Chandlee et al. (2015) examine the process of local iterative nasal spreading in Johore Malay, where contiguous sequences of vowels and glides are nasalized following a nasal. This function is depicted in Figure 5, where ‘N’ represents a nasal, ‘T’ any other consonant, ‘V-tilde’ a nasalized vowel or glide, and ‘V’ any other vowel or glide. Here, there are three distinct actions that arise from the letters: $\mathbb{1} = \langle 1, 2 \rangle$ from ‘V’, $x = \langle 1, 1 \rangle$ from ‘T’, and $y = \langle 2, 2 \rangle$ from ‘N’ and ‘V-tilde’. The letter ‘V’ is neutral because it does not change state, and so it corresponds to the identity action $\mathbb{1}$ (Lambert, 2023). The eggbox diagram revealing Green’s relations is shown in Figure 5.

This process is not definite, as there is an idempotent ($\mathbb{1}$) outside of the minimal \mathcal{J} -class. However, it is still \mathcal{L} -trivial: no two distinct elements are

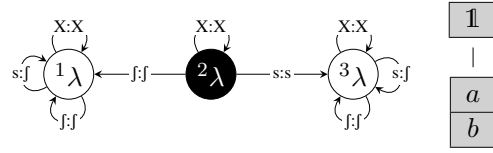


Figure 6: Symmetric harmony, with eggbox.

\mathcal{L} -related. It is still a band as well. However, if not for the neutral element it would be the *same* definite semigroup as the one witnessed for PNV. Algebraically, the semigroup satisfies the definition of tier-based definite. It belongs to $[[\mathbf{D}]]_T$.

The tier-based behavior can also be understood from the transducer. The state is determined by the most recent input symbol after projection to the tier $T = \{T, N, \tilde{V}\}$, as the suffixes $\{s, T\}$ lead to state 1 while $\{N, \tilde{V}\}$ lead to state 2. As mentioned, the only letter off the tier is ‘V’.

The right-to-left version of this process is not sequential, as a stream of ‘V’ must be buffered indefinitely to determine whether they must become ‘V-tilde’ or stay ‘V’, and so it shall not be analyzed.

This section has shown how processes of iterative spreading can be understood as a local process operating on a tier.

5.2 Symmetric Harmony

Heinz (2010) describes the symmetric harmony pattern of Navajo, where the existence of a [– anterior] sibilant such as ‘j’ triggers all prior [+ anterior] sibilants such as ‘s’ to assimilate and become [– anterior], and vice versa. The left-to-right version of this process is not sequential, as all sibilants must be buffered until the string ends to know which type surfaces. We therefore analyze only the right-to-left version, depicted in Figure 6, where ‘s’ represents a [+ anterior] sibilant, ‘f’ a [– anterior] sibilant, and ‘X’ any other segment.

There are three actions induced by the letters: $\mathbb{1} = \langle 1, 2, 3 \rangle$ from ‘X’, $a = \langle 1, 3, 3 \rangle$ from ‘s’, and $b = \langle 1, 1, 3 \rangle$ from ‘f’. Composition yields no new elements, and $\mathbb{1}$ is neutral.

The eggbox diagram is also shown in Figure 6. The semigroup is does not belong to \mathbf{D} because two elements are \mathcal{L} -related. On the other hand, no elements are \mathcal{R} -related, suggesting it may belong to \mathbf{K} . However, there is an idempotent outside the minimal \mathcal{J} -class and so it does not belong to \mathbf{K} . But it does satisfy Lambert’s (2023) definition of tier-based reverse definite. It belongs to $[[\mathbf{K}]]_T$.

Interestingly, symmetric harmony is the **dual**

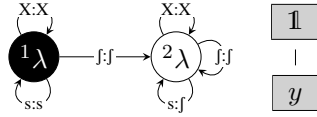


Figure 7: Asymmetric harmony, with eggbox.

of the structure for iterative spreading. For iterative spreading, the behavior is fixed by the k most recent symbols seen on the tier. For symmetric harmony, it is fixed instead by the *first* k symbols seen on the tier.

5.3 Asymmetric Harmony

Heinz (2010) also describes harmony in Sarcee, where only the [− anterior] sibilants are active. Again, this is not sequential when processing left-to-right, so we analyze only the right-to-left version. The minimal transducer is shown in Figure 7.

Two actions are induced by the letters: the neutral element $\mathbb{1} = \langle 1, 2 \rangle$ from ‘X’ and ‘s’, and $y = \langle 2, 2 \rangle$ from ‘f’. Composition yields no new elements. The eggbox is shown in Figure 7. As with iterative spreading, this is tier-based definite, and the degree of definiteness is one because it forms a band. The behavior is fixed by the most recently seen symbol on the tier. However, not only is it in $[\mathbf{D}]_T$ like iterative spreading, it is also in $[\mathbf{K}]_T$ like symmetric harmony because it is \mathcal{R} -trivial.

5.4 Discussion

The aforementioned analyses establish that canonical examples of output-oriented phonological processes belong to one or more of \mathbf{D} (definite), \mathbf{K} (reverse definite), $[\mathbf{D}]_T$ (the tier-based extension of definite), and $[\mathbf{K}]_T$ (the tier-based extension of reverse definite), when examining their left-to-right or right-to-left sequential transducers in minimal onward form. These results are summarized in Table 1. The class \mathbf{N} of semigroups are those which belong both \mathbf{D} and \mathbf{K} , and it is also a variety (Almeida, 1995).

These results are striking because the algebraic analysis groups local, iterative spreading together with non-local iterative spreading since they each invoke neutral elements (i.e. involve projections onto tiers). Our analyses here show that these canonical *output*-oriented processes are in some sense local, after projection to some tier, on the *input* side as well.

It is also of interest to consider the smallest algebraic variety which includes these classes. The

Pattern	→	←
Post-Nasal Voicing	\mathbf{D}	\mathbf{D}
Prog. Iterative Spreading	$[\mathbf{D}]_T$	–
Reg. Symmetric Harmony	–	$[\mathbf{K}]_T$
Reg. Asymmetric Harmony	–	$[\mathbf{N}]_T$
Pre-Nasal Voicing	\mathbf{D}	\mathbf{D}
Reg. Iterative Spreading	–	$[\mathbf{D}]_T$
Prog. Symmetric Harmony	$[\mathbf{K}]_T$	–
Prog. Asymmetric Harmony	$[\mathbf{N}]_T$	–

Table 1: Algebraic classification for left-to-right (→) and right-to-left (←) processing.

smallest variety containing both \mathbf{D} and \mathbf{K} is \mathbf{LI} , called “locally trivial” (Almeida, 1995) and sometimes “generalized definite” (Ginzburg, 1966; Brzozowski and Fich, 1984). Similarly, the tier-based extension of \mathbf{LI} , denoted $[\mathbf{LI}]_T$ contains $[\mathbf{D}]_T$ and $[\mathbf{K}]_T$. Interestingly, none of these tier-based extensions are varieties because it can be shown they are not closed under products (Lambert, 2022). We are thus motivated to identify the smallest variety which contains $[\mathbf{LI}]_T$.

Closing $[\mathbf{LI}]_T$ under products and quotients has one advantage: by definition, it necessarily includes processes that occur over *multiple* tiers. Recall that the tier T in the above classes is singular; consequently, the total phonology of those languages lies outside any one such class. For this reason, we call this closure \mathbf{MLI} , which can be read as “locally trivial over multiple tiers.” Almeida (1995) independently studied this class and others like it, considering \mathbf{M} as a natural operator linking varieties of semigroups with varieties of monoids.⁵

To sum up, the smallest algebraic variety which includes all the canonical phonological processes we have considered in this paper, as well as combinations thereof, is \mathbf{MLI} . Figure 8 shows the classes discussed in this paper, along with their containment relationships.

6 Inference

We examine the processes and their input-oriented analyses discussed above within the tradition of grammatical inference (de la Higuera, 2010; Heinz et al., 2015; Heinz and Sempere, 2016; Wieczorek, 2017). Specifically, we are interested in whether there is an algorithm which identifies those processes in the limit from positive data (Gold, 1967)

⁵A monoid is a semigroup with an identity.

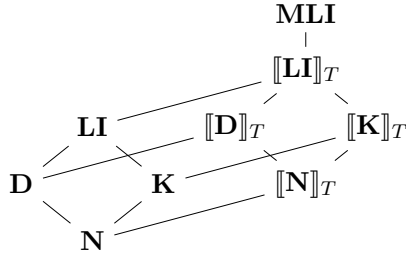


Figure 8: Containment among varieties and extensions.

in linear time and data (de la Higuera, 1997). Of course, they are learnable given their output-oriented structure (Chandlee et al., 2015), but relying only on the structure of the input can simplify the system.

Jardine et al. (2014) present SOSFIA, an algorithm which identifies in linear time and data any class of sequential functions representable with a single deterministic transducer. For any function in MLI , there is some k such that its behavior is fixed by the combinations of the first k symbols encountered and the most recent k symbols encountered, across all possible tiers. Consequently, we construct a family of learners, one for each value k . Each learning algorithm constructs a deterministic transducer for k - MLI , and then uses SOSFIA to determine the output of its edges.

Given k and a fixed input alphabet Σ , the construction begins by fixing the state space. The states are in one-to-one correspondence with contexts, where a context is the k first symbols (“prefixes”) and k most-recent symbols (“suffixes”) for each tier (i.e. all subsets of Σ). In cases where fewer than k salient symbols have been encountered, these shorter strings constitute the context. A prefix of length k is **saturated**. Starting with the initial state corresponding to the empty string on all tiers, expand the state space iteratively as follows until no new edges are created. For each newly created state q , consider the effect of appending a single letter $a \in \Sigma$: Saturated prefixes remain unchanged, as are unsaturated prefixes on tiers that exclude a , but unsaturated prefixes on tiers that include a are extended by appending a . Similarly, suffixes on tiers that exclude a remain unchanged, but suffixes on tiers that include a are extended by appending a and, if now longer than k , contracted by removing their initial symbol. The result is a state r . If r is a new state, then it is added to the state space. In any case, an edge is created from q to r whose input is a and whose output is \square , representing a blank.

Eventually, no new states will be created, and after the next iteration, no new edges will be created.

The state space is not small. There are $2^{|\Sigma|}$ possible tiers and more than $|\Sigma|^{2k}$ possible prefix–suffix pairs. Nonetheless, once the state space has been filled out, what remains is to assign outputs to the edges in a way that agrees with the observed data. This is precisely the problem SOSFIA solves (Jardine et al., 2014). Given a finite set of input–output pairs and an output-empty deterministic transducer as constructed above, this algorithm fills the outputs in such a way as to maintain onwardness.

If the sample contains sufficient information, which eventually it will in the identification in the limit paradigm, then all outputs will be filled. SOSFIA’s time and data complexities are linear, but the constant is large due to the enormous state space.

7 Conclusion

We examined Output (Tier-based) Strictly Local maps in concept and in practice. It was shown that no algebraic property can determine whether a process belongs to these classes (§4). We also provided algebraic analyses for a sample of linguistically relevant O(T)SL processes (§5). Of the processes considered, all lay in $[[\mathbf{D}]]_T$ or $[[\mathbf{K}]]_T$, with behaviors fixed either by the k most recent symbols or the first k symbols encountered, for some fixed k , after projection to some fixed tier T . Interestingly, all of the output-oriented maps we discussed were also bands, with all elements idempotent.

These algebraic analyses reveal the unfolding behaviors of these output-oriented functions in terms of their inputs. In particular, iterative spreading was shown to be a local process on tier, and only different from symmetric and asymmetric harmony with regards to whether the first or most recent symbols on the tier trigger harmony. These analyses recall the application of Rule 2 (Figure 1) and Walker’s (2014, p. 503) argument that “even in unbounded systems where harmony proceeds among adjacent vowels, the trigger-target relations may be nonlocal, with a single trigger related to many targets, both adjacent and nonadjacent.” One area for future linguistic research is a more extensive algebraic cataloging of local and long-distance phonological processes, with particular attention to any that lie outside of MLI (Jardine, 2016).

The third contribution was an instantiation of the SOSFIA inference algorithm (Jardine et al., 2014) in order to learn processes of the variety MLI in

the limit from positive samples (§6). While this is more powerful than necessary to capture the processes described in this work, it serves to demonstrate the learnability of the processes in question, even without relying on their Output (Tier-based) Strict Locality. Future research can examine imposing further restrictions to improve the space efficiency of the learning algorithm. Another important area of future research is to conduct a detailed comparison between this approach and others, such as the one in (Burness and McMullin, 2019) for 2-OTSL, and one for regular functions more generally (de la Higuera, 2010).

References

- 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, Salt Lake City, Utah.
- Jorge Almeida. 1995. *Finite Semigroups and Universal Algebra*, volume 3 of *Series in Algebra*. World Scientific, Singapore.
- Janusz Antoni Brzozowski and Faith Ellen Fich. 1984. [On generalized locally testable languages](#). *Discrete Mathematics*, 50:153–169.
- Phillip Burness and Kevin McMullin. 2019. [Efficient learning of output tier-based strictly 2-local functions](#). In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90, Toronto, Canada. Association for Computational Linguistics.
- Phillip Burness, Kevin McMullin, and Jane Chandlee. 2021. [Long-distance phonological processes as tier-based strictly local functions](#). *Glossa: a journal of general linguistics*, 6(1):1–37.
- 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.
- Christian Choffrut. 2003. [Minimizing subsequential transducers: A survey](#). *Theoretical Computer Science*, 292(1):131–143.
- Alfred Hombitzelle Clifford and Gordon Bamford Preston. 1961. *The Algebraic Theory of Semigroups*, volume 7 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, Rhode Island.
- Thomas Colcombet. 2011. [Green’s relations and their use in automata theory](#). In *Language and Automata Theory and Applications: Proceedings of the 5th International Conference, LATA 2011*, volume 6638 of *Theoretical Computer Science and General Issues*, pages 1–21, Heidelberg. Springer-Verlag.
- Colin de la Higuera. 1997. [Characteristic sets for polynomial grammatical inference](#). *Machine Learning*, 27(2):125–138.
- Colin de la Higuera. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.
- Samuel Eilenberg. 1976. *Automata, Languages, and Machines*, volume B. Academic Press, New York, New York.
- 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.
- Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. 2019. [Logical and algebraic characterizations of rational transductions](#). *Logical Methods in Computer Science*, 15(4):16:1–16:42.
- Abraham Ginzburg. 1966. [About some properties of definite, reverse-definite and related automata](#). *IEEE Transactions on Electronic Computers*, EC-15(5):806–810.
- Edward Mark Gold. 1967. [Language identification in the limit](#). *Information and Control*, 10(5):447–474.
- James Alexander Green. 1951. [On the structure of semigroups](#). *Annals of Mathematics*, 54(1):163–172.
- Gunnar Hansson. 2010. *Consonant Harmony: Long-Distance Interaction in Phonology*. Number 145 in University of California Publications in Linguistics. University of California Press, Berkeley, CA. Available on-line (free) at eScholarship.org.
- Jeffrey Heinz. 2010. [Learning long-distance phonotactics](#). *Linguistic Inquiry*, 41(4):623–661.
- Jeffrey Heinz, Colin de la Higuera, and Menno van Zaanen. 2015. *Grammatical Inference for Computational Linguistics*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.
- 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, 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: Short Papers*, volume 2, pages 58–64, Portland, Oregon. Association for Computational Linguistics.
- Jeffrey Heinz and José Sempere, editors. 2016. *Topics in Grammatical Inference*. Springer-Verlag, Berlin Heidelberg.
- Adam Jardine. 2016. [Computationally, tone is different](#). *Phonology*, 33(2):247–283.
- Adam Jardine, Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the Twelfth International Conference on Grammatical Inference*, volume 34 of *JMLR: Workshop and Conference Proceedings*, pages 94–108.
- Nancy C. Kula and Nasir A. Syed. 2020. Non-myopic nasal spreading in Saraiki. *Radical: A Journal of Phonology*, 1:126–182.
- Dakotah Lambert. 2022. *Unifying Classification Schemes for Languages and Processes with Attention to Locality and Relativizations Thereof*. Ph.D. thesis, Stony Brook University.
- Dakotah Lambert. 2023. [Relativized adjacency](#). *Journal of Logic, Language and Information*, 32(4):707–731.
- Dakotah Lambert and Jeffrey Heinz. 2023. [An algebraic characterization of total input strictly local functions](#). In *Proceedings of the Society for Computation in Linguistics*, volume 6, pages 25–34, Amherst, Massachusetts.
- Andrew Nevins. 2010. *Locality in Vowel Harmony*. Linguistic Inquiry Monographs. MIT Press, Cambridge, Massachusetts.
- Jean-Éric Pin. 1984. *Variétés de Langages Formels*. Masson, Paris.
- George Neal Raney. 1958. [Sequential functions](#). *Journal of the ACM*, 5(2):177–180.
- James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2012. [Cognitive and sub-regular complexity](#). In Glyn Morrill and Mark-Jan Nederhof, editors, *Formal Grammar 2012*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer-Verlag.
- Sharon Rose and Rachel Walker. 2004. [A typology of consonant agreement as correspondence](#). *Language*, 80(3):475–531.
- Marcel-Paul Schützenberger. 1977. [Sur une variante des fonctions séquentielles](#). *Theoretical Computer Science*, 4(1):47–57.
- Harry van der Hulst. 2018. *Asymmetries in Vowel Harmony*. Oxford University Press.
- Rachel Walker. 1998. *Nasalization, Neutral Segments, and Opacity Effects*. Ph.D. thesis, University of California, Santa Cruz.
- Rachel Walker. 2011. *Vowel Patterns in Language*. Cambridge University Press, Cambridge.
- Rachel Walker. 2014. Nonlocal trigger-target relations. *Linguistic Inquiry*, 45(3):501–523.
- Wojciech Wieczorek. 2017. *Grammatical Inference: Algorithms, Routines and Applications*. Springer.

Do speakers minimize dependency length during naturalistic dialogue?

Meghna Hooda Mudafia Zafar Samar Husain

Indian Institute of Technology Delhi

meghnahooda1706@gmail.com

mudafia@hss.iitd.ac.in

samar@hss.iitd.ac.in

Abstract

Dependency Length Minimization (DLM) is considered to be a linguistic universal governing word order variation cross-linguistically. However, evidence for DLM from large-scale corpus work is typically based on written (news) corpus and its effect on sentence production during naturalistic dialogue is largely unknown. Furthermore, Subject-Object-Verb languages are known to show a weaker preference for DLM. In this work, we test the validity of DLM using a dialogue corpus of Hindi, an SOV language. We also undertake a quantitative analysis of various syntactic phenomena that lead to DLM and compare the effect of DLM on both spoken and written modalities. Results provide novel evidence supporting a robust effect of DLM in spoken corpus. At the same time, compared to the written data, DLM was found to be weaker in dialogue. We discuss the implications of these findings on sentence production and on methodological issues with regards to the use of corpus data to investigate DLM.

1 Introduction

Understanding the structural complexity of natural language has been a key goal in psycholinguistics (e.g., Miller, 1962; Kimball, 1973; Hawkins, 1990; Levelt, 1972). This is because the formal properties of natural language can help us to uncover the underlying processes that subserve the generation and comprehension of such structures (Frazier, 1987; Levelt, 1989). These proposals are informed by our understanding of the severe resource constraint under which a dynamic system such as language comprehension/production operates (e.g., Just and Carpenter, 1992). An influential way to formalize complexity has been in terms of the arrangement of words in a sentence (Hudson, 1995; Wasow, 2002). On this account, called Dependency Length Minimization (DLM), two words that are syntactically related to each other would tend to appear in close

proximity rather than away from each other (Gibson, 1998, 2000). DLM can be understood in terms of optimizing limited memory resources – establishing a dependency relation between two words will typically require memory retrieval (of the head or the dependent), and these retrievals are known to be subject to locality considerations (Gibson, 1998; Lewis and Vasishth, 2005). This implies that sentences with shorter dependencies will, on average, be easier to process. Indeed, there is experimental evidence that an increase in dependency length leads to difficulty during comprehension as well as production (Grodner and Gibson, 2005; Bartek et al., 2011; Scontras et al., 2015).

Recent corpus-based work has provided strong cross-linguistic validation for DLM (Liu, 2008; Gildea and Temperley, 2010; Futrell et al., 2015; Temperley, 2007). These studies clearly demonstrate that DLM can be deemed as a linguistic universal across languages. If true, this has implications for the design properties of natural language and its architectural underpinnings (Futrell et al., 2020). However, a key issue with this claim is that corpus-based evidence for DLM mostly comes from written data (e.g., news genre). While both speaking and writing involve the same production apparatus, it is easy to see that they may not be operating under similar constraints. For instance, the production system can be assumed to be under more time pressure when speaking than writing, where it is typical to make many edits to a sentence (Biber, 2009; Hayes and Flower, 1986; Chafe, 1985). One reason for this is that the visual feedback during writing is more stable while the acoustic feedback during speech is momentary. Thus, it is reasonable to assume that the DLM constraint might be more evident in written text where the writer tries to achieve high readability for the reader. Meanwhile, in speech, other speaker-centric pressures related to incrementality, accessibility, etc., could supersede the DLM constraint (cf.

Levelt, 1989; Gleitman et al., 2007; Wheeldon and Konopka, 2023).

Another interesting finding in the literature is that while DLM operates cross-linguistically, it does not appear to be as strong across all languages. In particular, research suggests that the effect of DLM in Subject-Object-Verb (SOV) languages is less strong (Futrell et al., 2020; Dyer, 2023; Liu, 2020). Indeed, recent work using news data suggests that DLM has a marginal role in determining word order variation in an SOV language like Hindi (Ranjan et al., 2022). Thus, it is unclear if the DLM constraint would also hold in naturalistic spoken data in an SOV language, Hindi.

To summarize, there are two reasons to doubt the cross-linguistic generalizability of DLM: (a) large-scale validation of DLM has primarily been observed with written data, and (b) the effect of DLM has been observed to be weaker in SOV languages. In this work, we investigate if DLM is indeed operational in an SOV language Hindi during naturalistic dialogue. Further, if we do find evidence for DLM in the spoken modality, we are interested in probing the source of this effect. In particular, we investigate two well-known word order related phenomena that are known to be triggered by DLM, these are, the long-before-short pattern (Hawkins, 2014) and right-extrapolation (Wasow, 1997b). Finally, we compare the strength of DLM in the spoken vs written modality.

The paper is arranged as follows: in Section 2, we present our key experiment on investigating DLM using random baselines. In Section 3, we probe the results regarding DLM in dialogue corpus using two word order related phenomena. Following this, in Section 4, we compare the findings of the DLM experiment on dialogue corpus with written corpus. We consolidate all the findings and discuss their implications in Section 5. Section 6 concludes the paper.

2 DLM during dialogue

This section presents the key investigation of our work, i.e., can DLM be observed during naturalistic dialogue in an SOV language, Hindi? In order to test this question, we conduct a corpus-based study using the methodology proposed in Liu (2008); Futrell et al. (2015); Liu et al. (2017); Yadav et al. (2019). In particular, we compare real trees in a Hindi dialogue corpus with random baseline trees that match the real trees in certain formal properties.

The DLM distribution between these pairs of trees is compared to investigate the question at hand.

2.1 Data

The IIT Delhi Hindi Dialogue Corpus (Pareek et al., 2023) was used for the study. The dialogue data comprises of the Hindi segment of the CallFriend project (Canavan and George, 1996), which consists of 60 unscripted telephone conversations between Hindi native speakers. The spoken data was manually transcribed and later was (semi-)automatically annotated for part-of-speech and syntactic dependency relations. All annotations were finally validated manually. The current study is based on data comprising 31,020 sentences (mean sentence length = 6.13). For the purpose of this study, this dataset underwent a filtering process involving the exclusion of sentences containing code-switching, quotations, and incomprehensible content (i.e., words that were transcribed as *incomprehensible* because the audio was not clear). Non-lexical tokens such as laughter, pauses, and noise were also removed from the sentence. Finally, tokens representing disfluencies were also removed. This left us with a dataset comprising 28,953 sentences (mean sentence length = 5.68).

For generating the random baselines, we further subset this data to exclude sentences with lengths less than 3 and more than 19.¹ This gave us the final data comprising 22414 sentences that were used to generate the random baselines. The average sentence length in this data was 7.67.

2.2 Random Baselines

Following Yadav et al. (2022a,b), we generate a random baseline called random linear arrangement baselines (RLAs) for the real dependency trees obtained from the Hindi Dialogue Corpus. The algorithm chooses a random baseline tree from a uniform distribution of random linearization of a real tree through a rejection sampling method. The random tree is controlled for sentence length, the number of crossing dependencies and all topological properties (e.g., node arity, tree depth, etc.). Critically, the baseline preserves the dependency relations of a real tree. This makes RLAs a relatively strict baseline compared to simple random

¹This was necessitated because the compute time to generate the conservative baselines for sentences more than 19 was very high. Sentences with less than 3 words were removed because the random baselines generated for such sentences remain invariant. Note that the sentence length was computed by excluding the punctuation.

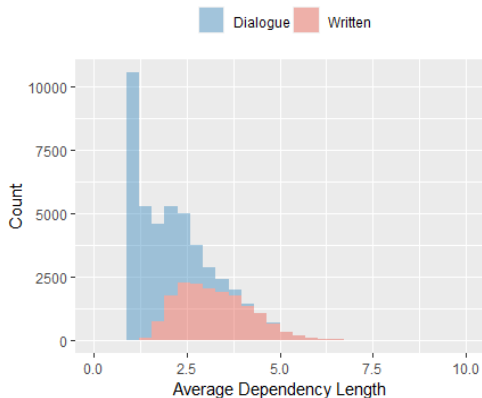


Figure 1: Distribution of Average Dependency Length in Dialogue and Text Corpus

structure baselines where the tree topology is not controlled (cf. Futrell et al., 2015). A random tree is generated for each real tree. We then compare the two trees to test if the average dependency length between the two trees differs (for more details see Yadav et al., 2022b).

2.3 Statistical Method

In order to investigate if DLM is indeed observed during a naturalistic speech in Hindi, we test whether the distribution of dependency length is significantly different between real trees and the baseline trees. Dependency length was computed as the number of words intervening between the head and its dependent. We compare the growth of average dependency length with sentence length in real vs random trees. If DLM holds in speech data, then compared to random baselines, this growth in real trees should be slower. We fit a linear mixed-effects model using the *lme4* (Bates et al., 2015) package in R (R Core Team, 2022) with dependency length as the dependent variable, sentence length, and tree type (real or random) as fixed effects, and Interlocutor-pair as the random effect (see Eq 1). The random effect captures the variation across different speakers in the dataset. The key coefficient of interest in the model is the interaction between sentence length and tree type. Note that in treating the interaction as the coefficient of interest we follow Ferrer-i Cancho and Liu (2013); Futrell et al. (2015); Gildea and Temperley (2010) who show that dependency length should be considered as a function of sentence length. This is because the effect of capturing the average difference of dependency length between real and random trees

could be inaccurate, as dependencies could come from varying sentence lengths.

$$DL \sim \text{Sentence.length} * \text{Tree.Type} + (\text{Sentence.length} * \text{Tree.Type} | \text{Interlocutor} - \text{pair}) \quad (1)$$

Maximal models were fit, subject to model convergence (Barr et al., 2013).

2.4 Results

Table 1 shows the results of the linear mixed model analysis. Results show that the average dependency length grows slower with sentence length in real trees compared to random trees (p-value < 0.001). This can also be visually observed in Figure 2. These results show that DLM is observed in the dialogue corpus.

Table 1: Results from the linear mixed models. Tree.Type (Random vs Real) was coded as treatment contrast with the random tree as the baseline. Sentence length (SL) was scaled.

	estimate	SE	t-value	Pr(> t)
Intercept	2.34	0.005	397.79	<0.001
SL	0.54	0.005	104.97	<0.001
Real	-0.36	0.01	-31.99	<0.001
SL:Real	-0.17	0.008	-21.08	<0.001

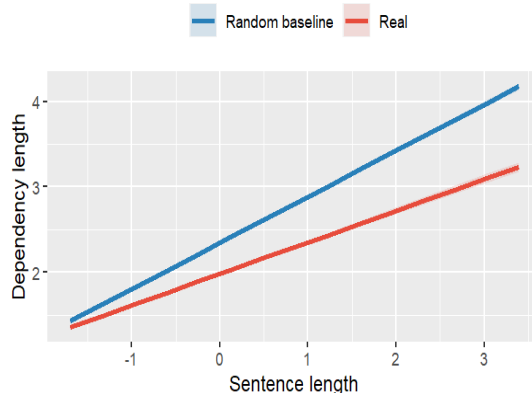


Figure 2: Fitted models showing the growth of dependency length with respect to sentence length in real language trees compared to Random Linear Arrangement (RLA) baselines.

3 What drives DLM?

Large-scale corpus based investigations are important because they help us in testing the ecological validity and generalizability of a theory like DLM.

At the same time, we also need to uncover the underlying cause for the observed results in terms of various syntactic configurations. One such configuration concerns the length of a constituent – the length of a constituent has been argued to guide word order changes that minimize the overall dependency length in that utterance (Hawkins, 1994, 2004, 2014). With regard to SOV languages, a long-before-short order of constituents can be deemed to lessen the overall dependency length compared to a short-before-long order. This can schematically be seen in Figure 3.

Evidence for a long-before-short preference has been found from production experiments in SOV languages (Yamashita and Chang, 2001; Ros et al., 2015; Faghiri and Samvelian, 2020). For example, in Japanese sentences 1a-1b, consisting of a short subject and a long object, Yamashita and Chang (2001) found that speakers produced more non-canonical (OSV) order sentences like 1b than the canonical (SOV) order like 1a.

- (1) a. [_S keezi-ga] [_O se-ga takakute
 detective-NOM height-NOM tall
 gassiri sita hannin-o]
 and big-boned suspect-ACC
 [_V oikaketa]
 chased
 ‘The detective chased the suspect who
 is tall and big-boned.’
- b. [_O se-ga takakute gassiri sita hannin-o]
 [_S keezi-ga] [_V oikaketa]

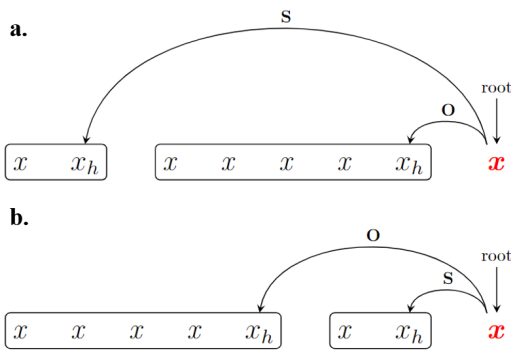


Figure 3: Two possible ordering patterns for a short subject (S) and a long object (O) in SOV languages. In (a) the short S is placed first while in (b), the long O is placed first. Order (b) has a shorter dependency length than (a).

The other configuration that has been argued to be driven by DLM is right-extrapolation (Hawkins,

2014; Wasow, 1997b; Arnold et al., 2000; Wasow and Arnold, 2003; Szmrecsányi, 2004; Yngve, 1960; Gibson, 1998).² In the context of SOV languages, this would imply placing the phrase in question *after* the clause final verb. Here we explore two such configurations, one where the noun is modified by a relative clause (Kothari, 2010; Zafar and Husain, 2023) (see, Example 2a, 2b), and the other where the noun is modified by a non-verbal phrase (e.g., another noun phrase; Example 3a, 3b).

3.1 Long-before-short order

In the previous section we demonstrated DLM in the Hindi Dialogue corpus. In this section we ask if the DLM effect is driven by a long-before-short pattern. We do this by examining the word order patterns for core arguments such as subjects and objects;³ in addition, we also investigate this effect for adjuncts. In particular, we investigate if increase in object/adjunct length will increase a shift from canonical to non-canonical OSV/AdjSV order to align with a long-before-short pattern and whether this shift leads to reduced average dependency length of the utterance. The key prediction is that preverbally, a long argument/adjunct should be placed before a short argument/adjunct when this leads to reduced dependency length.

3.1.1 Method

We extracted SOV and OSV utterances from the dialogue corpus. While doing so, we ensured that the sentences had (a) only S and O as the two core arguments, (b) both arguments were dependent on the same verbal head, (c) both the arguments preceded the verb, and (d) did not involve any crossings. For computing phrasal length, case-markers were considered as part of their respective noun. We obtain 1152 SOV instances and 274 OSV instances for the analysis.

Generalized linear models with the logit function (Nelder and Wedderburn, 1972) were fit to the data. The order of arguments (SOV or OSV) was the dependent variable, and phrasal length was the independent variable consisting of 3 levels: Equal; Subject Long and Object Long. The Equal condition served as the baseline because DLM would

²Explanations for right-extrapolation have traditionally been cast in terms of phrasal length or complexity (e.g., Yngve, 1960; Wasow, 2002) However, this point is not relevant for the current discussion as right-extrapolation due to increased phrasal length leads to DLM (see, Yadav et al., 2022a).

³We did not include ditransitive sentences as part of the analysis because they were less in number.

not prefer one order over the other in this condition. Scaled average dependency length was also added as an additional predictor (see Equation 2). The effect of Object Long, as well as its interaction with average dependency length, formed the coefficients of interest – this is because while the effect of Object long captures the shift due to length, the interaction tells us if this shift leads to DLM. Specifically, if the long-before-short effect exists in dialogue then we expect the likelihood of shifts from the SOV to the OSV order to increase when the Object is long compared to when both the Subject and the Object are of equal length. This would mean that the coefficient of Object Long should have a positive sign. Additionally, the interaction of dependency length with Object Long should have a negative sign. This would tell us that the likelihood of shifts to the OSV order in the O Long condition decreases as the dependency length of the sentence increases. This implies that higher OSV shifts when the Object is long correspond to lower DLM.

$$\text{Order}(SOV|OSV) \sim (\text{Subject.Long} + \text{Object.Long}) * \text{Avg.Dependency.Length} \quad (2)$$

We additionally investigated shifts from the Subject Adjunct Verb (SAdjV) patterns to the Adjunct Subject Verb (AdjSV) pattern as a function of length. This analysis helps us investigate if the long-before-long order exists irrespective of the nature of the verbal modification, i.e., argument or adjunct. Using the criterion mentioned for SOV sentences previously, 699 SAdjV instances and 644 AdjSV instances were extracted for analysis. All sentences had only one core argument – the subject, and one adjunct. The *glm* model for this analysis is shown in Equation 3 and is similar to the analysis we ran for argument shifts.

$$\text{Order}(SAdjV|AdjSV) \sim (\text{Subject.Long} + \text{Adjunct.Long}) * \text{Avg.Dependency.Length} \quad (3)$$

3.1.2 Results

Table 2 shows the results. With regard to the SOV/OSV analysis, we find that compared to the Equal condition, in sentences with long objects, the tendency to place the object initially (leading to a

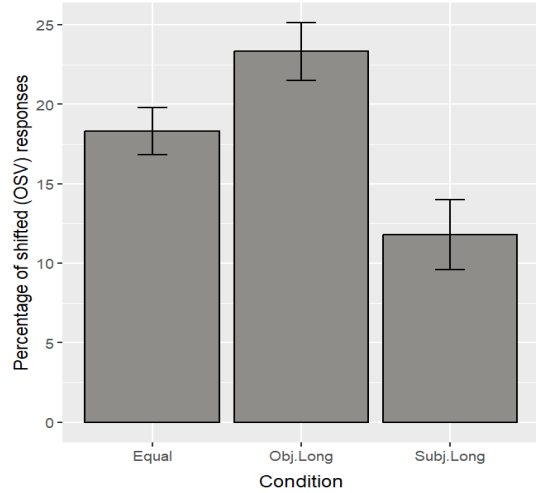


Figure 4: Percentage of Object-fronted responses in the Equal, Subject-Long and Object-Long conditions in the Dialogue Corpus.

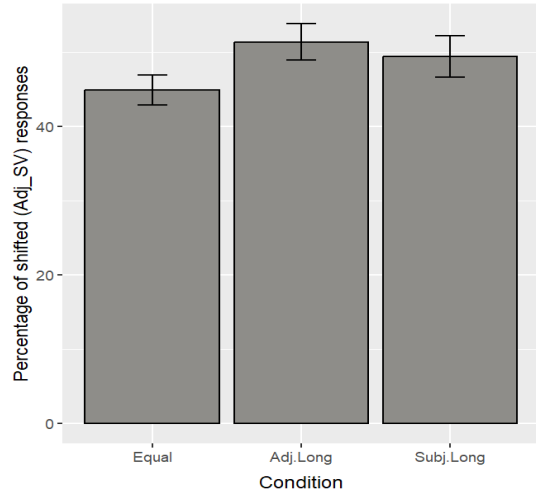


Figure 5: Percentage of Adj-fronted responses in the Equal, Subject-Long and Adj-Long conditions in the Dialogue Corpus.

long-before-short OSV order) increases ($p=0.01$) (also see Figure 4). In addition, we also observe a significant interaction between object length and average dependency length ($p=0.03$) such that compared to the equal length subject/object, the proportion of OSV order decreases when the average dependency length increases. An opposite trend was observed for utterances with long subjects suggesting that the tendency to shift decreased when the subject was longer than the object.⁴

⁴A careful reader will observe that the effect of Avg.DL in the models is not really meaningful for the discussion at hand; it represent the effect of dependency length in the utterances

A similar long-before-short pattern was found in the SAdjV/AdjSV utterances. Long adjuncts were fronted to form an AdjSV more when the adjunct was longer than the subject compared to when they were of equal length ($p=0.009$) (see Figure 5). In addition, there was a significant interaction between the length of the adjunct and average dependency length ($p<0.001$) such that the tendency to form an AdjSV utterance reduced with increased dependency length. Broadly, the above results show that during dialogue, speakers follow a long-before-short pattern for DLM.

3.2 Right-extrapolation

The long-before-short word order configuration discussed above leads to reduced dependency length through changes in word order preverbally. However, as discussed earlier, in certain configurations dependency length in SOV languages can also be minimized by placing a long phrase *after* the matrix verb via right extrapolation. Below we report the analysis for two types of right extrapolations, one where the noun is modified by a relative clause (Kothari, 2010; Zafar and Husain, 2023) (see, Example 2b), and the other where the noun is modified by a non-verbal phrase (e.g., another noun phrase; Example 3b).

3.2.1 Method

We began by extracting all instances of nominal modifications from the dialogue corpus. These (clausal or non-clausal) modifiers could either appear *in-situ* or could be right-extrapolated post-verbally.⁵ This gave us 199 right-extrapolated utterances and 1710 *in-situ* utterances. See examples 2a and 3a for clausal and non-clausal *in-situ* modifications, respectively; Examples 2b and 3b show their right-extrapolated counterparts.

- (2) a. vo vali job **jo kar rahey**
 DEM PART job REL do PROG
thein chor diye
 be.PST.PRF leave give.PST.PRF
 ‘I left that job which I was doing’
- b. vo vali job chor diye **jo kar rahey**
thein
- (3) a. aur **tumhare dushman ki** shadi
 and your enemy POSS marriage
 ho gayi
 be.PRES go.PST.PRF

where the length of both subject and object is equal.

⁵For clausal modifications *in-situ* implies a post-nominal modification; while for non-clausal modifications *in-situ* means a pre-nominal modification.

‘And your enemy got married?’

- b. aur shadi hogayi **tumhare dushman ki**

Similar to section 3.1, we ran a generalized linear model with the logit link function where the order (Right-extrapolated or In-situ) was the dependent variable, and scaled phrasal length was the independent variable. Scaled average dependency length was added as an additional predictor (see Equation 4). If right-extrapolation is driven by dependency length minimization, then we should observe an effect of phrasal length such that as phrasal length increases, right-extrapolation should increase, i.e., we should observe a positive sign on the coefficient. In addition, we ought to also observe a negative coefficient for the interaction between phrasal length and dependency length – this will suggest that shifts to the right-extrapolated order are less likely when such shifts increase the total dependency length of the sentence. Together the two effects would imply that right-extrapolation for long phrases correspond to lower DLM.

$$Order(RightExtrapolated|InSitu) \sim PhrasalLength * DependencyLength \quad (4)$$

3.2.2 Results

Table 3 shows the results for the glm analysis. The key finding was that the shift to the right-extrapolated order increased with an increase in phrasal length ($p<0.001$). However, the interaction between phrasal length and dependency length was also positive. This means that the shift from *in-situ* to right-extrapolation in fact increased with an increase in dependency length for higher values of phrasal length ($p=0.01$). Thus, the results suggest that the increase in right-extrapolation for long phrases in dialogue is not driven by DLM.

4 DLM in speech vs written text

As discussed in Section 1, a key difference between dialogue and written text concerns the time window under which the final utterance is produced. Typically, the time available to produce an utterance (such as the ones found in the current dialogue corpus) during naturalistic dialogue will be much less than the time taken to produce an edited sentence in written corpus. Indeed, it is well known that turn-taking during dialogue is very fast (Clark, 2014). This suggests that DLM, which will require considerable resources due to planning, could be more

Table 2: Word Order analysis for the long-before-short experiment in Section 3.1. Treatment contrast was used in both models (Equal length argument/adjunct formed the baseline); ARG = Arguments, ADJ = Adjuncts, Avg.DL = Average Dependency Length, Subj.Long = Subject longer than Object, Obj.Long = Object longer than Subject, Adj.Long = Adjunct longer than Subject. Avg.DL was scaled before fitting the model. Significant effects where p-value<0.05 have been highlighted.

	Dialogue				Written				
	estimate	SE	z-value	p-value	estimate	SE	z-value	p-value	
ARG	Intercept	-1.59	0.09	-17.2	< 0.001	-2.6	0.06	-38.1	< 0.001
	Subj.Long	-0.61	0.26	-2.36	0.001	-0.27	0.18	-1.51	0.12
	Obj.Long	0.36	0.14	2.49	0.01	-0.11	0.15	-0.71	0.47
	Avg.DL	0.06	0.08	0.78	0.43	-0.26	0.07	-3.8	0.0001
	Subj.Long:Avg.DL	0.80	0.21	3.69	< 0.001	0.13	0.18	0.74	0.45
	Obj.Long:Avg.DL	-0.32	0.15	-2.07	0.03	-0.42	0.16	-2.53	< 0.01
ADJ	Dialogue				Written				
	estimate	SE	z-value	p-value	estimate	SE	z-value	p-value	
	Intercept	-0.02	0.57	-0.33	0.73	0.05	0.03	1.43	0.15
	Subj.Long	0.18	0.13	1.32	0.18	0.3	0.09	3.3	< 0.001
	Adj.Long	0.34	0.13	2.58	0.009	-0.03	0.09	-0.4	0.68
	Avg.DL	-0.05	0.05	-0.93	0.35	-0.01	0.03	-0.38	0.69
	Subj.Long:Avg.DL	0.08	0.14	0.62	0.53	0.28	0.09	3	0.002
Adj.Long:Avg.DL	-0.54	0.13	-4.02	< 0.001	-0.32	0.09	-3.49	< 0.001	

Table 3: Results from the glm models for right extraposition in dialogue and written. Treatment contrast was used in the model. DL: Dependency Length and PL: Phrasal Length. Significant effects where p-value<0.05 have been highlighted.

	Dialogue				Written			
	estimate	SE	t-value	p-value	estimate	SE	t-value	p-value
Intercept	-2.27	0.08	-28.09	< 0.001	-4.29	0.05	-74.136	< 0.001
PL	0.5	0.06	7.53	< 0.001	0.97	0.02	37.13	< 0.001
DL	-0.03	0.08	-0.41	0.67	-0.02	0.056	-0.52	0.6
PL:DL	0.16	0.06	2.39	0.01	-0.1	0.01	-5.67	0.01

visible in written corpus while speaker-centric factors such as accessibility, etc., could be more dominant in speech production (cf. Arnold et al., 2000; MacDonald, 2013; Ferreira and Dell, 2000). Below we investigate this possibility.

4.1 Method

To investigate the strength of DLM in dialogue and written data, we follow the method discussed in Section 2. Similar to the experiment for the dialogue corpus, RLA random baselines were used for comparison. To do this comparison, we needed a baseline that can be compared with both dialogue as well as written text trees. So, we select those sentences in the dialogue and written data that match in three critical topological features – sentence length, max arity, and max tree depth. This enables us to generate an RLA baseline, which can be com-

pared with real trees from two modalities. Using this criterion, we got 869 triplets of baseline and dialogue/written trees.

$$Dependency.Length \sim Sentence.length * Tree.Type \quad (5)$$

A linear model was used to investigate the increase in dependency length with respect to sentence length in random vs real trees of dialogue and written text (Equation 5). As before, the key coefficient (which should be negative) will be the 2-way interactions between real trees in dialogue/written text and sentence length.

4.2 Results

Table 4 shows the results. The results show a significant interaction between sentence length with

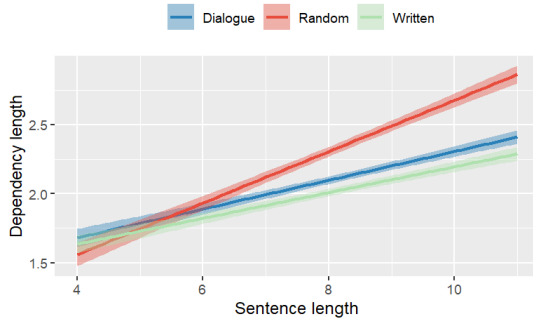


Figure 6: Fitted models showing the growth of dependency length with respect to sentence length in real trees from dialogue and written data compared to RLAs.

Table 4: Estimates from the fitted linear models comparing DL in real and RLAs baseline tree for the two modalities for the effect of dependency length(scaled). SL = sentence length.

	Estimate	SE	t-value	p-value
Intercept	2.32	0.01	151.23	<0.001
SL	0.35	0.01	23.07	<0.001
Real _D	-0.21	0.02	-9.99	<0.001
Real _W	-0.3	0.02	-14.15	<0.001
SL:Real _D	-0.15	0.02	-7.18	<0.001
SL:Real _W	-0.17	0.02	-8.14	<0.001

both dialogue trees ($p < 0.001$) and written trees ($p < 0.001$). This shows that DLM is being minimized in both written and dialogue data when compared to the common baseline.⁶ Interestingly, the effect-size of the coefficient suggests that DLM is comparatively stronger in written data compared to the dialogue data. This can be clearly seen in Figure 6.

In additional analyses, we also investigated if the effects discussed in Section 3.1 and 3.2 can be observed in written data using the HTDB corpus. Results show that, unlike in dialogue, we do not find evidence for a long-before-short pattern in the written data (no significant increase in fronting when the object or the adjunct was long) in both SOV ($p = 0.47$) as well as in the SAdjV utterances ($p = 0.68$). Interestingly, the data showed evidence for DLM in the case of right-extrapolation

⁶Similar to the dialogue data analysis, we also tested for DLM in written data independently using the entire written corpus (Bhatt et al., 2009). For analysis, sentences with length more than 2 and less than 12 were used. As expected, DLM is minimized in Hindi written data (Futrell et al., 2015; Dyer, 2023).

($p < 0.001$). The details of these analyses can be found in Tables 2, 3.

5 Discussion

The current paper provides novel evidence in support of DLM in a dialogue data for an SOV language, Hindi. We investigated two phenomena that are implicated in DLM, namely, long-before-short and right-extrapolation. Our results show that while long-before-short leads to DLM in the dialogue data, right-extrapolation does not.

With regard to the comparison between dialogue and written data, while we find evidence for DLM induced long-before-short pattern in the dialogue data, we did not find any evidence for this in the written data (see Table 2). On the other hand, while we find evidence for DLM induced right-extrapolation in the written data, we did not find such evidence in the dialogue data (see Table 3). To probe this further, we investigated the modifier type in right-extrapolated situations in both dialogue and written data. We find that right-extrapolation is dominated by clausal modifiers in written data – in the dialogue data, only 21% of right-extrapolated modifiers are clausal; while this was 85% in the written data. It is known that such clausal extractions lead to DLM in Hindi (see, Zafar and Husain, 2023). In addition, in such configurations, the average length for clausal modifiers in dialogue was 6.65 words, while it was 10.8 words in the written data. Similarly, the average length for right-extrapolated modifiers (clausal and non-clausal) was longer in written (11.1 words) than in dialogue (3.7 words) (cf. Biber, 2009). This shows that right-extrapolation is an important DLM strategy in written data but not in dialogue. One might ask, if right-extrapolation is not motivated by DLM, why do we find increased shifting with an increase in phrasal length in dialogue? This trend for right-extrapolation of long phrases after the verb could be due to other reasons such as information structure (Butt and King, 1996; Huck and Na, 1990), ease of planning (Wasow, 1997a) or syntactic expectation (Levy et al., 2012).

The fact that DLM is minimized more in written data than dialogue data (cf. Table 4) is consistent with not only the fact that written data is a product of an extensive editing process, but also that the writing process itself can be very different from speaking (Wengelin et al., 2009); also see, Roeser et al. (2019). The comparative analysis of written

vs dialogue data is also quite instructive from a methodological perspective. Corpus-based investigations on DLM link their findings to the underlying cognitive processes (e.g., Futrell et al., 2015). The current work shows that while DLM can be observed in these different modalities, the underlying causes for the manifestation of DLM might be quite different. Therefore, any such generalizations should also be based on the syntactic configurations that lead to DLM. For example, there could be other causes to DLM in addition to the ones explored here, e.g., elision (Kramer, 2021).

The DLM constraint in the dialogue data has implications for production models that assume incrementality (Levelt, 1989). In particular, DLM minimization implies that speakers would have to structurally plan some components of the utterance before articulating them. This would mean that planning during language production is non-incremental to a certain degree (cf. Wheeldon and Konopka, 2023). At the same time, these results also highlight certain constraints on planning scope. The results suggest that, in dialogue, speakers do not plan very long post-nominal modifiers.

Together, these results highlight the over-arching influence of working-memory constraints on production process in an SOV language like Hindi (cf. Slevc, 2011; Gennari et al., 2012; Humphreys et al., 2016). Future work needs to investigate how such constraints interacts with other factors such as accessibility (cf. Ranjan et al., 2022).

6 Conclusion

In a corpus-based investigation, we test the generalizability of DLM as a cognitive principle for word order variation in a Hindi naturalistic spoken data. Our results show that the real trees attested in a dialogue corpus of Hindi have on average shorter dependencies when compared to random trees that match the real trees in topological features. Furthermore, to understand the sources of DLM in dialogue, we zoom into two phenomena known to minimize dependency length. We find that DLM in dialogue is primarily minimized by fronting longer arguments and adjuncts and not by right extraposing clausal or nominal modifiers. Finally, we compare the strength of DLM in spoken and written data. We posit that DLM is minimized in both the modalities, its effect being stronger for written than spoken. Overall, these results shed light on the overarching influence of working memory con-

straints in governing syntactic choices during both language comprehension and production.

Acknowledgments

We would like to thank the anonymous reviewers whose insightful comments contributed to enhancing the quality of the paper. Sincere thanks to Himanshu Yadav for comments on the early draft of the paper and for offering valuable suggestions regarding the analysis presented in Section 4. We are also grateful to the team involved in the development of the IIT Delhi Hindi Dialogue Corpus for providing the data. This work was partially funded by a grant from the Department of Science and Technology’s Cognitive Science Research Initiative (Grant no. DST/CSRI/2018/69) to Samar Husain. The project details and code used in this work is available at <https://osf.io/vxd6w/> and <https://github.com/MeghnaHooda/DLM-in-naturalistic-dialogue.git>.

References

- Jennifer E. Arnold, Anthony Losongco, Thomas Wasow, and Ryan Ginstrom. 2000. Heaviness vs. newness: The effects of structural complexity and discourse status on constituent ordering. *Language*, 76(1):28–55.
- 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.
- Brian Bartek, Richard L Lewis, Shravan Vasishth, and Mason R Smith. 2011. In search of on-line locality effects in sentence comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(5):1178.
- Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*.
- Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *proceedings of the third linguistic annotation workshop (LAW III)*, pages 186–189.
- Douglas Biber. 2009. Are there linguistic consequences of literacy? comparing the potentials of language use in speech and writing. *Cambridge handbook of literacy*, pages 75–91.
- Miriam Butt and Tracy Holloway King. 1996. Structural topic and focus without movement. *Online Proceedings of LFG*.

- Alexandra Canavan and Zipperlen George. 1996. CALLFRIEND Hindi LDC96S52. *Web Download*. Philadelphia: Linguistic Data Consortium.
- Wallace Chafe. 1985. Linguistic differences produced by differences between speaking and writing. *Literacy, language, and learning: The nature and consequences of reading and writing*, 105:105–123.
- Herbert H Clark. 2014. 18 spontaneous discourse. *The Oxford Handbook of Language Production*, page 292.
- Andrew Thomas Dyer. 2023. Revisiting dependency length and intervener complexity minimisation on a parallel corpus in 35 languages. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 110–119.
- Pegah Faghiri and Pollet Samvelian. 2020. Word order preferences and the effect of phrasal length in SOV languages: evidence from sentence production in Persian. *Glossa: a journal of general linguistics*.
- Victor S Ferreira and Gary S Dell. 2000. Effect of ambiguity and lexical availability on syntactic and lexical production. *Cognitive psychology*, 40(4):296–340.
- Ramon Ferrer-i Cancho and Haitao Liu. 2013. The risks of mixing dependency lengths from sequences of different length. *arXiv preprint arXiv:1304.3841*.
- Lyn Frazier. 1987. **Syntactic processing: Evidence from dutch**. *Natural Language & Linguistic Theory*, 5(4):519–559.
- Richard Futrell, Roger P Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96(2):371–412.
- Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.
- Silvia P. Gennari, Jelena Mirković, and Maryellen C. MacDonald. 2012. Animacy and competition in relative clause production: A cross-linguistic investigation. *Cognitive Psychology*, 65(2):141 – 176.
- Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.
- Edward Gibson. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, pages 95–126.
- Daniel Gildea and David Temperley. 2010. Do grammars minimize dependency length? *Cognitive Science*, 34(2):286–310.
- Lila R Gleitman, David January, Rebecca Nappa, and John C Trueswell. 2007. On the give and take between event apprehension and utterance formulation. *Journal of memory and language*, 57(4):544–569.
- Daniel Grodner and Edward Gibson. 2005. Consequences of the serial nature of linguistic input for sentential complexity. *Cognitive science*, 29(2):261–290.
- John A Hawkins. 1990. A parsing theory of word order universals. *Linguistic inquiry*, 21(2):223–261.
- John A Hawkins. 1994. *A performance theory of order and constituency*. 73. Cambridge University Press.
- John A Hawkins. 2004. *Efficiency and complexity in grammars*. OUP Oxford.
- John A Hawkins. 2014. *Cross-linguistic variation and efficiency*. OUP Oxford.
- John R Hayes and Linda S Flower. 1986. Writing research and the writer. *American psychologist*, 41(10):1106.
- Geoffrey J Huck and Younghee Na. 1990. Extraposition and focus. *Language*, pages 51–77.
- Richard Hudson. 1995. Measuring syntactic difficulty. *Manuscript, University College, London*.
- Gina F. Humphreys, Jelena Mirković, and Silvia P. Gennari. 2016. Similarity-based competition in relative clause production and comprehension. *Journal of Memory and Language*, 89:200 – 221. Speaking and Listening: Relationships Between Language Production and Comprehension.
- Marcel A Just and Patricia A Carpenter. 1992. A capacity theory of comprehension: individual differences in working memory. *Psychological review*, 99(1):122.
- John Kimball. 1973. Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47.
- Anubha Kothari. 2010. *Processing constraints and word order variation in Hindi relative clauses*. Ph.D. thesis, Stanford University.
- Alex Kramer. 2021. Dependency lengths in speech and writing: A cross-linguistic comparison via youdepp, a pipeline for scraping and parsing youtube captions. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 359–365.
- Willem JM Levelt. 1972. Some psychological aspects of linguistic data. *Linguistische Berichte*, 17:18–30.
- Willem JM Levelt. 1989. *Speaking: From intention to articulation*. Cambridge: The MIT Press.
- Roger Levy, Evelina Fedorenko, Mara Breen, and Edward Gibson. 2012. The processing of extraposed structures in english. *Cognition*, 122(1):12–36.
- Richard L Lewis and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3):375–419.

- Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191.
- Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: a new perspective on syntactic patterns in natural languages. *Physics of life reviews*, 21:171–193.
- Zoey Liu. 2020. Mixed evidence for crosslinguistic dependency length minimization. *STUF-Language Typology and Universals*, 73(4):605–633.
- Maryellen C. MacDonald. 2013. How language production shapes language form and comprehension. *Frontiers in Psychology*, 4:226.
- George A Miller. 1962. Some psychological studies of grammar. *American psychologist*, 17(11):748.
- John Ashworth Nelder and Robert WM Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 135(3):370–384.
- Benu Pareek, Mudafia Zafar, Karan Yadav, Meghna Hooda, Ashwini Vaidya, and Samar Husain. 2023. Introducing the hindi dialogue corpus. In *Poster presented at the fourth South Asian Forum on the Acquisition and Processing of Language*. <https://osf.io/76akt>.
- R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Sidharth Ranjan, Rajakrishnan Rajkumar, and Sumeet Agarwal. 2022. Linguistic complexity and planning effects on word duration in hindi read aloud speech. *Proceedings of the Society for Computation in Linguistics*, 5(1):119–132.
- Jens Roeser, Mark Torrance, and Thom Baguley. 2019. Advance planning in written and spoken sentence production. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 45(11):1983.
- Idoia Ros, Mikel Santesteban, Kumiko Fukumura, and Itziar Laka. 2015. Aiming at shorter dependencies: the role of agreement morphology. *Language, Cognition and Neuroscience*, 30(9):1156–1174.
- Gregory Scontras, William Badecker, Lisa Shank, Eunice Lim, and Evelina Fedorenko. 2015. Syntactic complexity effects in sentence production. *Cognitive science*, 39(3):559–583.
- L Robert Slevc. 2011. Saying what’s on your mind: working memory effects on sentence production. *Journal of experimental psychology: Learning, memory, and cognition*, 37(6):1503.
- Benedikt Szmrecsányi. 2004. On operationalizing syntactic complexity. In *7th International Conference on Textual Data Statistical Analysis*, pages 1032–1039. Presses Universitaires de Louvain, Louvain-la-Neuve.
- David Temperley. 2007. Minimization of dependency length in written english. *Cognition*, 105(2):300–333.
- Thomas Wasow. 1997a. End-weight from the speaker’s perspective. *Journal of Psycholinguistic research*, 26:347–361.
- Thomas Wasow. 1997b. Remarks on grammatical weight. *Language Variation and Change*, 9(1):81–105.
- Thomas Wasow. 2002. *Postverbal Behaviour*. CSLI Publications.
- Thomas Wasow and Jennifer Arnold. 2003. Post-verbal constituent ordering in english. *Topics in English Linguistics*, 43:119–154.
- Åsa Wengelin, Mark Torrance, Kenneth Holmqvist, Sol Simpson, David Galbraith, Victoria Johansson, and Roger Johansson. 2009. Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior Research Methods*, 41(2):337–351.
- Linda Ruth Wheeldon and Agnieszka Konopka. 2023. *Grammatical Encoding for Speech Production*. Elements in Psycholinguistics. Cambridge University Press.
- Himanshu Yadav, Samar Husain, and Richard Futrell. 2019. Are formal restrictions on crossing dependencies epiphenominal? In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 2–12.
- Himanshu Yadav, Samar Husain, and Richard Futrell. 2022a. Assessing Corpus Evidence for Formal and Psycholinguistic Constraints on Nonprojectivity. *Computational Linguistics*, 48(2):375–401.
- Himanshu Yadav, Shubham Mittal, and Samar Husain. 2022b. A reappraisal of dependency length minimization as a linguistic universal. *Open Mind*, 6:147–168.
- Hiroko Yamashita and Franklin Chang. 2001. “Long before short” preference in the production of a head-final language. *Cognition*, 81(2):B45–B55.
- Victor H Yngve. 1960. A model and an hypothesis for language structure. *Proceedings of the American philosophical society*, 104(5):444–466.
- Mudafia Zafar and Samar Husain. 2023. Dependency locality influences word order during production in sov languages: Evidence from hindi. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45.

Do language models capture implied discourse meanings? An investigation with exhaustivity implicatures of Korean morphology

Hagyeong Shin

Department of Linguistics
University of California San Diego
hashin@ucsd.edu

Sean Trott

Department of Cognitive Science
University of California San Diego
sttrott@ucsd.edu

Abstract

Markedness in natural language is often associated with non-literal meanings in discourse. Differential Object Marking (DOM) in Korean is one instance of this phenomenon, where post-positional markers are selected based on both the semantic features of the noun phrases and the discourse features that are orthogonal to the semantic features. Previous work has shown that distributional models of language recover certain semantic features of words—do these models capture implied discourse-level meanings as well? We evaluate whether a set of large language models are capable of associating discourse meanings with different object markings in Korean. Results suggest that discourse meanings of a grammatical marker can be more challenging to encode than that of a discourse marker.

1 Introduction

The distributional hypothesis states that the meaning of a word can be derived in part from the linguistic contexts in which it is used (Harris, 1954). Accordingly, language models built on distributional patterns of language use have proved useful for modeling both content words (Lenci, 2018; Boleda, 2020) and function words (Marelli and Baroni, 2015). With recent advancements, large scale neural language models have demonstrated that distributional semantic models encode substantial amount of semantic knowledge (Tenney et al., 2019a; Trott and Bergen, 2021) and even contextualized meanings of words (Tenney et al., 2019b). In the current work, we ask whether distributional semantics in a large scale can encode meanings of function words that encompasses semantic and discourse meanings.

Differential Object Marking (DOM), particularly in Korean, poses an interesting challenge to distributional semantics. DOM is a phenomenon in which grammatical objects can be marked with different grammatical structures (e.g., case markers)

(see Bossong, 1991), often explained with varying semantic features of the referent (see Aissen, 2003). In Korean, DOM is associated not only with the semantic features of an object but also with the *discourse status* of the object (Kwon and Zribi-Hertz, 2008; Lee, 2006). As an instance, three alternative markings—*lul*, *nun*,¹ and null-marking—can appear after an object that is picked out from a set of contextualized alternatives. While the null-marking option reflects that the object is contextualized element in the discourse context, *lul* and *nun* markers imply the *exhaustive status* of the object, contrasted with other alternatives from the set. Furthermore, each of the *lul* and *nun* markers derives the exhaustive status that posit different constraint on the upcoming discourse.

Thus, in order to account for Korean DOM, distributional patterns of object markings must be able to encode multiple meanings of the markers at once: the discourse features that are grounded with the discourse context, as well as semantic features of objects that are orthogonal to discourse context. To test this, we used pre-trained Large Language Models (LLMs). LLMs have been evaluated as a subject of psycholinguistics (Futrell et al., 2019) for their ability to grasp non-literal meanings (Hu et al., 2023; Jeretič et al., 2020) and those that are closely tied to the pragmatic context (Trott et al., 2023). In line with these studies, we evaluate whether LLMs exhibit the competency with discourse meanings associated with different object markings in Korean.

2 Patterns under evaluation

In this study, we focus on three post-positional marking options in Korean, as illustrated in (1).²

¹The *lul* and *nun* markers are allomorphic and are respectively realized as *ul* and *un* to follow a syllable that ends with a coda (consonant). For clarity, we refer to each marker as *lul* and *nun*.

²Following abbreviations are used in the examples. \emptyset = null-marking, ACC = accusative, ADD = additive, CT = con-

The first *lul* marker is canonically an accusative case marker. Without any considerations on specific discourse context, the *lul* marker indicates that a noun phrase is a grammatical direct object. The *nun* marker, on the other hand, is mainly used as a discourse marker and replaces the *lul* marker. In (1), the *nun* marker indicates that the object is being contrasted with other entities (Choi, 1996; Kim, 2018). The last marking option in (1), annotated with \emptyset , indicates that no markers follow the object. This option, what we refer to as “null-marking”, is common in colloquial Korean (Choi-Jonin, 2008; Lee, 2007). All of the three marking options in (1) are grammatical and felicitous to appear after an object.

- (1) Mina-ka pizza-**{lul/nun/ \emptyset }** kacyewasse.
Mina-NOM pizza-**{ACC/CT/ \emptyset }** brought
'Mina brought a pizza.'

Some options listed in (1), however, can lead to *implicatures* in a discourse context. Implicatures refer to the meaning that is not conveyed by the truth-conditional meaning of each words, but what can be inferred from those in the context of the usage (Grice, 1989). As a specific type of implicature, *exhaustivity implicature* arises when there is a set of alternatives in the discourse. When one element from the set is picked out as an answer to the question, the mentioned element is perceived as an exhaustive information relevant to the question (Büring, 2003; Horn, 1981; Rooth, 1992; Van Rooij and Schulz, 2004).

The *lul* and *nun* marker can both evoke exhaustivity implicatures, while the null-marking does not. To illustrate, in (2), the given context evokes a set of alternatives $\{pizza, cake\}$. In the first response option (A), one alternative *pizza* from the set is picked out but appears without any marker. In this response, the null-marking does not convey an exhaustivity status of the object, but indicates that the speaker refers to the contextualized object in the discourse context.

- (2) Context: Interlocutors know that guests were invited to bring a pizza and a cake to the party, and that Mina attended the party.
- Q. What did Mina bring?
- A. Pizza- \emptyset kacyewasse.
pizza- \emptyset brought
'(Mina) brought the pizza.'

trastive, NOM = nominative, NEG = negation, # = infelicitous.

- B. Pizza-**lul** kacyewasse. Cake-to
pizza-ACC brought cake-ADD
kacyewasse.
brought
'(Mina) brought the pizza.
(\rightarrow ~~Mina didn't bring anything else.~~)
(Mina) also brought the cake.'
- C. Pizza-**nun** kacyewasse. #Cake-to
pizza-CT brought cake-ADD
kacyewasse.
brought
'(Mina) brought the pizza.
(\rightarrow Mina didn't bring anything else.)
(Mina) also brought the cake.'
- C'. Pizza-**nun** kacyewasse. Cake-un ahn
pizza-CT brought cake-CT NEG
kacyewasse.
brought
'(Mina) brought the pizza.
(\rightarrow Mina didn't bring anything else.)
(Mina) didn't bring the cake.'

Both of the *lul* and *nun* markers can evoke exhaustivity implicatures, but each marker's implicatures differ in terms of whether the information conveyed by the implicature can be corrected in the upcoming discourse. In other words, the *lul* and *nun* marker's exhaustivity implicatures exhibit different *cancelability* (Lee, 2003, 2017). In (2) B, *pizza* is picked out from the set of alternatives and marked with the *lul*. The *lul* marker conveys an exhaustivity implicature (given in the parentheses), which is cancelable in the upcoming discourse (Lee, 2003). Due to its cancelability, the second utterance '(Mina) also brought a cake.' forms a felicitous continuation of the response.

C and C' in (2) illustrates the exhaustivity implicature evoked by the *nun* marker. However, unlike the *lul* marker, the *nun* marker evokes an exhaustivity implicature that is not cancelable (Kim, 2018; Lee, 2003). In C, the second sentence cannot be the felicitous discourse continuation, because it contradicts the uncancelable exhaustivity implicature derived by the *nun* marker. C' presents the felicitous discourse continuation, where the exhaustivity status of the object indicated by the *nun* is not contradicted.³

³To be more precise, exhaustive interpretations of the *lul* and *nun* markers have been discussed in association with different types of Information Structure (Lambrecht, 1994). The *lul* marker is associated with the Contrastive Focus (CF) status and derives the exhaustivity of the CF element(s) (Lee, 2003, 2017). The *nun* marker is associated with the Contrastive Topic (CT) status, and implies that there is unanswered portion of a Question Under Discussion (QUD) (Büring, 2003).

Thus, paradigmatic contrasts between *lul*, *nun*, and null-marking derive non-literal meanings that are captured in the discourse domain: exhaustivity implicatures and their cancelability in association with each marking options. In the following sections, we examine whether LLMs’ semantic representations of each marking options reflect these discourse meanings.

3 Models under evaluation

To assess distributional semantics with discourse meanings of Korean DOM, we examine a set of pre-trained LLMs. Pre-trained LLMs are trained to perform a token prediction task on large volumes of corpora (e.g., sometimes billions or trillions of words) using many parameters. LLMs learn to predict words in context by observing statistical patterns in which words and word sequences are most likely to co-occur. This makes them well-suited as operationalizations of the distributional hypothesis.

Previous research suggests that distributional semantics may be able to effectively encode discourse meanings, particularly when the training data and the model are both sufficiently large. Tenney et al. (2019b) discovered that more contextualized knowledge may emerge in deeper layers of a model’s architecture. Jeretič et al. (2020) found that models are adept at learning non-literal meanings, even though off-the-shelf models may lack pragmatic competency. Additionally, Hu et al. (2023) observed that larger models achieve high accuracy and align with human error patterns in various pragmatic phenomena. Considering these results, we evaluate models with different numbers of parameters to determine whether distributional semantics can effectively encode discourse meanings associated with different object markings as the models scale up.

We test a series of generative pre-trained transformer models that are developed for the Korean language and as multi-lingual models. Starting from models with smaller parameters to larger ones, we include KoGPT-2 (125M)⁴ and KoGPT-Trinity (1.2B),⁵ both developed and trained specifically for Korean. We also test Polyglot-Ko mod-

els with 3.8B,⁶ 5.8B,⁷ and 12.8B⁸ parameters, developed under a project for multilingual LLMs and primarily trained with Korean data. Lastly, we test text-davinci-003/GPT-3 (175B)⁹ and gpt4-1106-preview/ChatGPT, both accessed with OpenAI API.¹⁰ These two models stand out as they are trained with reinforcement learning from human feedback (RLHF); this makes them less suitable to a direct test of the “pure” distributional hypothesis (given that they receive explicit human feedback), but they remain useful operationalizations of what can be learned from a linguistic training signal. Despite not being developed specifically for Korean, their massive size and diverse training data enable them to demonstrate competency in using the Korean language.

All models, with the exception of ChatGPT, grant access to their internal semantic representations through logits/log probabilities assigned to input and output tokens. ChatGPT, on the other hand, is optimized for ‘prompting,’ limiting the assessment of its semantic representation via the model’s meta-judgments on inputs. We employ distinct approaches to evaluate these two model types.

4 Experiment 1: Discourse meanings in processing

We first investigate whether LLMs’ semantic representation of words can reflect discourse meanings of the *lul* and *nun*. We compare LLMs’ and humans’ sensitivity towards exhaustivity implicatures indicated with the *lul* and *nun* markers and the different cancelability of each marker’s implicatures.

We created 288 stimuli (48 items to appear in 6 conditions), as shown in Table 1. Each item begins with a sentence contextualizing a set of alternatives, followed by a question about one alternative from the set. The response portion of the conversation always consists of two sentences. The first sentence was manipulated to have different object markings between *lul* and *nun*. The second sentence states how the other alternative from the set forms the relation with the elided subject. The previous verb from the question (and the first response sentence)

⁶<https://huggingface.co/EleutherAI/polyglot-ko-3.8b>

⁷<https://huggingface.co/EleutherAI/polyglot-ko-5.8b>

⁸<https://huggingface.co/EleutherAI/polyglot-ko-12.8b>

⁹Accessed before its deprecation on January 4th, 2024.

¹⁰<https://platform.openai.com/docs/api-reference>

To identify discourse patterns that language models can be evaluated for, we defer discussions on theoretical notions.

⁴<https://huggingface.co/skt/kogpt2-base-v2>

⁵<https://huggingface.co/skt/ko-gpt-trinity-1.2B-v0.5>

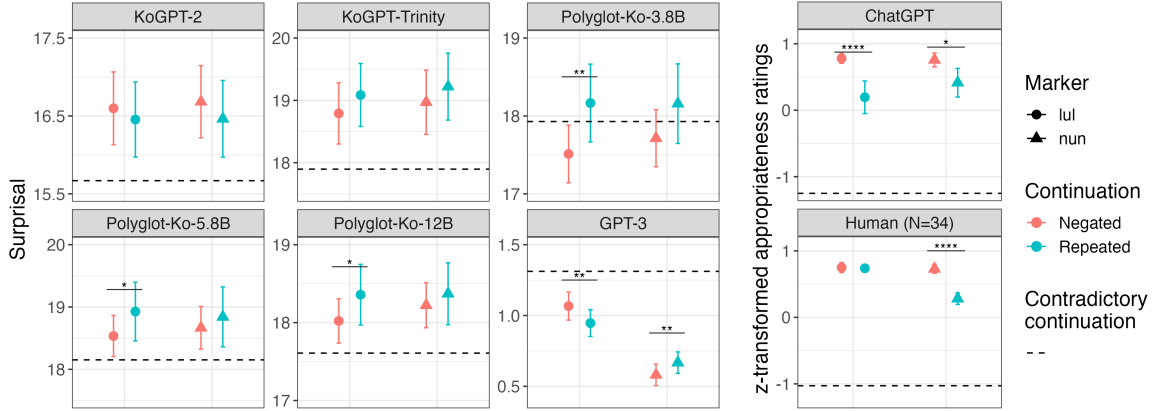


Figure 1: Evaluations of discourse continuations where the exhaustivity status implied by *lul* and *nun* is canceled (with repeated continuations) or not canceled (with negated continuations), and where the previous sentence is logically contradicted (dashed line). Mean surprisals and 95% CIs gathered from the 6 models are presented on the left. Higher surprisals indicate that the model had lower expectations for encountering the continuation. On the right side, mean of z-transformed appropriateness ratings and 95% CIs from ChatGPT and 34 native Korean speakers are presented. Higher ratings indicate that the discourse continuation was evaluated as more felicitous. Stars indicate adjusted significance levels obtained from paired *t*-tests with Bonferroni corrections (****: $p < 0.001$, ***: $p < 0.005$, **: $p < 0.05$).

<i>Set of alternatives</i>	Sohee knows that Mina and Yuna could receive medals and trophies.
<i>Question</i>	Sohee: What did Yuna receive?
<i>Object</i>	Mina: Received a medal- {lul/nun} .
<i>Continuation</i>	{Also received/Didn't receive/Only received} a trophy.

Table 1: An illustration of items used in the Experiment 1. Non-italic components in the right column are presented in Korean, and only one option in curly brackets is presented in a single item. See the Appendix A for this example written in Korean.

is manipulated to be either repeated or negated. When the previous verb is repeated, the exhaustivity status established in the previous sentence is canceled; when it is negated, the exhaustivity status is maintained. As a baseline, we also included contradictory continuations where the alternative object (e.g., *trophy* in Table 1) is marked with the ‘only’ (*-man*) marker and evokes a logical contradiction with the first sentence.

4.1 Surprisal measurements

As an assessment of semantic representations of different marking options, we obtained surprisal of each sentence following different markings (e.g.,

Continuation sentence in Table 1) from all models except ChatGPT. Logits assigned to each token in the sentence are first converted into log probabilities. We then summed log probabilities of all tokens in a sentence and normalized for the number of tokens in a sentence. These are then converted to surprisal. For GPT-3, we skipped the step of converting logits into log probabilities, as the output already provides log probabilities. If models perceive the exhaustivity implicature and its cancelability, they should exhibit notably higher surprisal only when the *nun* marker is followed by a repeated verb continuation (e.g., “also received” in Table 1).

Surprisal scores are summarized in the left sub-figure of Figure 1. Among the six models, only the GPT-3 could distinguish logical contradiction (dashed line in Figure 1) from pragmatically infelicitous discourse. Additionally, GPT-3 was the only model sensitive to the non-cancelable exhaustivity implicature of the *nun* marker. However, the model exhibited higher surprisal when encountering an uncanceled implicature following the *lul* marker, deviating from the observed pattern in (2) B and from human speakers. All Polyglot-Ko models showed sensitivity towards the *lul* marker followed by non-canceled exhaustivity, which is also incoherent with the cancelability observed in (2) B and from human speakers.

Coefficients	β of surprisal (<i>p</i> -value)						β of z-transformed ratings (<i>p</i> -value)	
	KoGPT-2	Ko-Trinity	Poly-3B	Poly-5B	Poly-12B	GPT-3	ChatGPT	Human
Intercept	16.60 (***)	18.79 (***)	17.51 (***)	18.53 (***)	18.02 (***)	1.07 (***)	0.78 (***)	0.75 (***)
Marker:Nun	0.08 (0.46)	0.18 (0.30)	0.20 (0.15)	0.13 (0.24)	0.20 (*)	-0.49 (***)	-0.02 (0.85)	-0.02 (0.65)
Continuation:Repeated	-1.14 (0.21)	0.30 (0.09)	0.65 (***)	0.39 (***)	0.34 (***)	-0.12 (***)	-0.58 (***)	-0.01 (0.88)
Marker:Nun × Continuation:Repeated	-0.08 (0.64)	-0.05 (0.85)	-0.21 (0.29)	-0.22 (0.17)	-0.19 (0.14)	0.21 (***)	0.24 (0.15)	-0.44 (***)

Table 2: Surprisal scores obtained with each of the first six models are fitted with the mixed effects model: $\text{surprisal} \sim \text{marker} * \text{continuation} + (1|\text{item})$. Positive coefficients indicate increase in the surprisal, thus decrease in the model’s expectedness. Ratings obtained with ChatGPT and human are z-transformed and fitted with the mixed effects model. For ChatGPT: $\text{z-rating} \sim \text{marker} * \text{continuation} + (1|\text{item})$. For human: $\text{z-rating} \sim \text{marker} * \text{continuation} + (1+\text{marker}*\text{continuation}|\text{participant}) + (1+\text{marker}*\text{continuation}|\text{item})$. In these models, positive coefficients indicate increase in the ratings of appropriateness. ***: $p < 0.001$, *: $p < 0.05$.

4.2 Elicited ratings

ChatGPT’s and humans’ processing patterns were assessed by asking them to rate how much each type of continuations (c.f. Table 1) is appropriate to follow the previous sentence. A 7-point likert scale was provided, with 1 representing the second sentence being ‘inappropriate’ and 7 representing it being ‘appropriate’. For ChatGPT, we set a system message—*Make sure to respond only with a number between 1 and 7.*—via OpenAI API. This can be considered as a meta-instruction guiding how the model should respond to each prompt. We also set the temperature to 0, which guides the model to generate responses deterministically based on the probability assignments. For human participants, the context preamble was written more specifically to describe the shared knowledge of speakers and listeners regarding the set of alternatives (see the Appendix A for the full item). In the human experiment, contradictory continuations were presented as a part of fillers. Each participant saw 24 critical items appearing in one out of 4 conditions, 24 fillers, and 4 attention checks. Native speakers of Korean were recruited and compensated via online crowdsourcing platform based in South Korea.¹¹ After the data elimination process, responses from 34 participants are retained and reported.

We transformed ChatGPT’s and each of the 34 participant’s appropriateness ratings, including ratings on contradictory continuations, into z-scores, using the mean and standard deviation obtained within each participant/model. The z-transformed ratings are presented in the right subfigure in Figure 1. Raw ratings without z-transformation are

presented in the Appendix B. When interpreting ChatGPT’s results, we adopt a cautious and non-conclusive approach, guided by the findings of Hu and Levy (2023). Instead of viewing the chat responses as a direct reflection of the model’s semantic representation, we consider them as indicative of the model’s proficiency in making evaluations about the input.

In general, ChatGPT and humans showed positive ratings for critical items targeting appropriateness ratings, while giving negative ratings for contradictory continuations targeting truth-conditional judgements (dashed line in Figure 1). This confirms that their ratings were based on discourse (in)felicity, not on truth-conditional meanings. ChatGPT showed similar sensitivity observed with GPT-3. The model was somewhat sensitive towards the *nun* marker’s non-cancelable exhaustivity implicature, which was coherent with human speakers’ patterns. However, the model also produced more negative ratings for the *lul* marker’s cancelable implicature, which differed from the pattern observed in human speakers.

4.3 Mixed-effects models

We fitted a mixed effects regression model to further investigate each model’s sensitivity towards the cancelability of the exhaustivity implicature. For models assessed with surprisal measurements, we fitted a model predicting surprisal with marker, continuation, and the interaction of the two, while controlling for random effects of each lexicalized item. For ChatGPT and humans assessed with ratings, we fitted a model predicting z-transformed ratings. For predicting humans’ ratings, random effects of each participant were additionally con-

¹¹<https://gosurveasy.com/>

trolled. Results are summarized in Table 2, with models listed from smaller to larger ones.

Comparing the first two KoGPT series model with Polyglot-Ko models, we observe that the Polyglot-Ko models have gained sensitivity towards the repeated continuation. As the Polyglot-Ko models reaches 12B parameters, they also gain sensitivity towards the markedness of an object. GPT-3 exhibited sensitivity to the infelicity arising from the *nun* marker followed by canceled exhaustivity (Marker:Nun×Continuation:Repeated). Additionally, it displayed sensitivity to other factors that human speakers did not perceive as influencing discourse felicity. ChatGPT demonstrated sensitivity only to the continuation factor that human speakers were insensitive to.

5 Experiment 2: Discourse meanings in production

From the previous experiment, we observed that Polyglot-Ko models and GPT-3 were surprised to see the *lul* marker followed by the canceled exhaustivity, while human speakers accepted the *lul* marker followed by either canceled or non-canceled exhaustivity. As noted with the example (1) in Section 2, the *lul* marker is canonically a grammatical case marker, and its grammatical function persists without the considerations on discourse context. Although the experimental items were designed to evoke the *lul* marker’s discourse function, it needs to be confirmed that the results from Experiment 1 stem from models and humans associating the *lul* marker with its additional discourse meaning (exhaustivity implicature), not solely from its grammatical function (marking the grammatical objects). To address this, we conducted the second experiment.

We designed 480 stimuli (48 items to appear with 10 manipulated components) exemplified in

<i>Intended message</i>	Mina intends to respond that Yuna received {only the medal/both the medal and the trophy} .
<i>Question</i>	Sohee: What did Yuna receive?
<i>Response</i>	Medal- {lul/nun/∅} received.

Table 3: An example of items used in the Experiment 2. Non-italic components in the right column are presented in Korean, and only one option among others in curly brackets are presented in the actual item. See the Appendix C for the item written in Korean.

Table 3. Each item started with the *intended message*, wherein the exhaustivity status of an object that a speaker intends to indicate was manipulated. Then, a question on the object and the response with three different marking options followed. The response sentence could have *lul*-, *nun*-, or null-marked object. As a baseline, we included a ‘contradictory’ response that does not match the (non-)exhaustivity status described in the intended message. Additionally, a ‘verbatim repeat’ response was included, maintaining the exact structure of the relative clause from the intended message statement. If LLMs can indeed associate the *lul* and *nun* markers with the exhaustivity status of the object, they should generate *lul*-marked and *nun*-marked responses when exhaustivity of the object is intended, and null-marked responses when exhaustivity of the object is not intended.

5.1 Probability measurements

Assessing all models except ChatGPT, we measured log probabilities assigned to the response sentences with different object markings. Log probability of a sentence is obtained in the same manner as in Experiment 1: logits of each token in a sentence are converted into log probabilities, which are then summed and normalized for the number of tokens in a sentence; only with GPT-3, log probabilities are directly accessed. We report the log probabilities instead of surprisal for this experiment in order to directly reflect the likelihood of generating different object markings.

We created two different types of prompts from one item, one including the intended message statement and one without it—and subtracted the log probabilities obtained from the former from the log probabilities obtained from the latter. This was to show how much the exclusivity status in the intended message is associated with the log probabilities assigned to each response type. In other words, models saw 960 prompts, with 480 containing the intended message statement and 480 without it. We report 480 log probability measurements obtained from each pair from each model.

Results are reported in Figure 2. Except GPT-3, all models struggled to assign higher probability to verbatim responses than to contradictory responses. Excluding this baseline results, Polyglot-Ko-12B demonstrated some competency in associating the *nun* marker with exhaustivity status: It assigned lower probabilities to *nun*-marked responses when

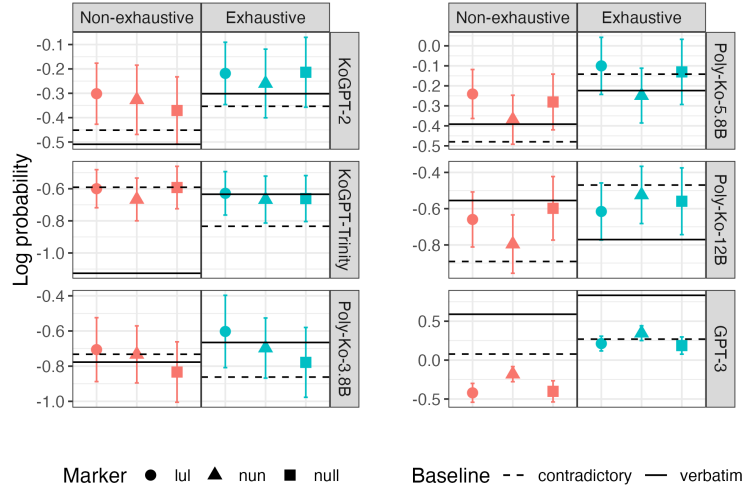


Figure 2: Mean log probabilities assigned to *lul*-, *nun*-, and null-marked responses when non-exhaustive or exhaustive messages are intended are shown. Error bars indicate 95% CIs. Dashed horizontal lines indicate the mean log probabilities assigned to contradictory responses, such as “Received only the medal” when a non-exhaustive message (both the medal and the trophy) is intended, or “Received both the trophy and the medal” when an exhaustive message (only the medal) is intended. Solid horizontal lines indicate the mean log probabilities assigned to verbatim responses, such as “Received both the trophy and the medal” when a non-exhaustive message (both the medal and the trophy) is intended, or “Received only the trophy” when an exhaustive message (only the medal) is intended.

	β of log probability (<i>p</i> -value)					
	KoGPT-2	Ko-Trinity	Poly-3.8B	Poly-5.8B	Poly-12B	GPT-3
Intercept	-0.59 (***)	-0.83 (***)	-0.37 (***)	-0.28 (***)	-0.60 (***)	-0.40 (***)
Exhaustivity:Yes	-0.07 (0.14)	0.05 (0.30)	0.16 (***)	0.15 (***)	0.04 (0.28)	0.59 (***)
Marker: <i>lul</i>	-0.01 (0.88)	0.13 (*)	0.07 (0.08)	0.04 (0.34)	-0.06 (0.09)	-0.02 (0.54)
Marker: <i>nun</i>	-0.07 (0.11)	0.10 (0.06)	0.04 (0.26)	-0.09 (*)	-0.20 (***)	0.22 (***)
Exhaustivity:Yes×Marker: <i>lul</i>	0.04 (0.55)	0.05 (0.51)	-0.07 (0.18)	-0.01 (0.87)	0.00 (0.92)	0.04 (0.32)
Exhaustivity:Yes×Marker: <i>nun</i>	0.07 (0.31)	-0.02 (0.80)	-0.09 (0.10)	-0.03 (0.62)	0.23 (***)	-0.06 (0.17)

Table 4: Log probabilities obtained with the six models are fit with the mixed effects model: $\log \text{ probability} \sim \text{exhaustivity} * \text{marker} + (1 | \text{item})$. Baseline model is set as Exhaustivity:No, Marker:Null. Positive coefficients indicate that the models assign higher probability to a response in varying exhaustivity in the intended message. ***, $p < 0.001$. *, $p < 0.05$.

exhaustivity was not intended, and higher probabilities when exhaustivity was intended. In the first experiment, we observed that GPT-3 exhibited processing patterns that were partially comparable to humans. However, GPT-3 did not associate the intended exhaustivity provided in the prompt with different marking options, as it did not assign significantly higher probabilities to *lul* or *nun*-marked responses when exhaustivity needed to be delivered.

5.2 Mixed-effects models

Probability measurements obtained from the six models in Figure 2 are further analyzed with a set of mixed-effects models. We fitted a mixed-effects model predicting log probability of a response sentence with the intended exhaustivity status of the

object and different markings of the object, while controlling random effects of each lexicalization of an item. Results are reported in Table 4.

Seen with the Polyglot-Ko-3.8B, Polyglot-Ko-5.8B, and GPT-3, larger models are more likely to assign distinct probabilities when intended exhaustivity differs. Although not in the direction that matches patterns in natural language, larger models (Polyglot-Ko-5.8B, Polyglot-Ko-12B, GPT-3) appear at least to gain sensitivity towards paradigmatic selections of marking options, as they assigned significantly different probabilities to *nun*-marked responses. Again, the results with Polyglot-Ko-12B is notable, as it assigns significantly higher probability to *nun*-marked responses when exhaustivity needs to be delivered.

No models associated intended exhaustivity with

the *lul* marker. Since the *lul* marker is canonically a grammatical case marker, this indicates that encoding dual meanings of a marker may be more challenging to LLMs. Considering this result with the Experiment 1 (c.f., Figure 1), we conclude that significantly different surprisals observed with the *lul* marker in Experiment 1 are unlikely to have come from associating the marker with the exhaustivity implicature. Rather, it is likely that the models’ sensitivity towards verb continuations were heightened when more canonical object marker (*lul*) appeared.

5.3 Forced-choice responses

We elicited responses from ChatGPT and humans with forced-choice tasks. Forced-choice tasks had the *intended message*, the *question* portion (c.f. Table 3), and either a pair of *lul*-marked and null-marked response, or a pair of *nun*-marked and null-marked as ‘response sets’. By presenting the options in these pairs, we aimed to ensure that participants and the model did not select the *lul* marker solely based on its grammatical function. If participants or the model associates *lul* and *nun* with exhaustivity implicatures but not with null-markings, the expected choices would be *lul* over null-marked responses and *nun* over null-marked responses.

After providing the response sets, we asked to choose the best message among the two to send as a response to the preceding question. For ChatGPT, we set the temperature to 0 and provided the system message via OpenAI API: *Please choose the response as you would speak in everyday conversations. Provided options may not express everything that you need to say. Nevertheless, please choose the best option among the two.* ChatGPT was presented with one prompt twice, once each with switched order of the response. In total, ChatGPT was presented with 192 prompts (48 items with 2 response sets, twice with switched order of response options), all in a zero-shot manner.

In the human experiment, the order of the marked and unmarked options of responses were randomly switched in every question. Each participant saw 24 critical items, 24 fillers on subject markings, and 4 attention check items. Human participants are also given the instruction, after each question, to choose the most proper response even if none of the two can represent everything that the speaker needs to say. See the Appendix C for the forced-choice tasks written in Korean. Participants were recruited and compensated via online

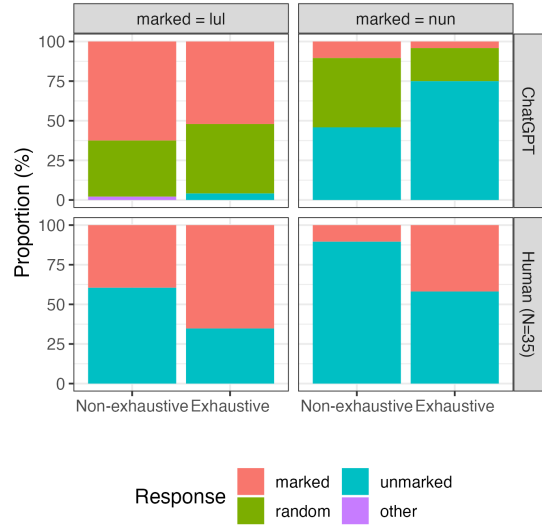


Figure 3: Proportions of responses elicited from ChatGPT and from 35 human participants. The left panels (marked = *lul*) summarize choices when the response sets included *lul*-marked and null-marked objects. Here, the ‘marked’ proportion, colored in red, indicates the proportion of *lul*-marked responses, while the ‘unmarked’ proportion, in blue, indicates the proportion of null-marked responses. On the right panels (marked = *nun*), choices are summarized when response sets included *nun*-marked and null-marked objects. Here, the ‘marked’ proportion, colored in red, indicates the proportion of *nun*-marked responses, while the ‘unmarked’ proportion, in blue, indicates the proportion of null-marked responses.

crowdsourcing platform based in South Korea.¹² After the data elimination process, we report 35 participants’ responses.

Results of forced-choice responses are summarized in Figure 3. To begin, ChatGPT frequently made ‘random’ choices, meaning that the model didn’t select the same option when the order of two responses was flipped. Even when disregarding the random choices, the model exhibited a pattern contrary to that found in human responses. When exhaustivity needed to be expressed, ChatGPT more often selected null-marked responses, whereas humans were more likely to mark the object with either the *lul* or *nun* marker. Overall, ChatGPT appears incompetent at generating the *lul* or *nun* markers to evoke the implicature in our tasks. This result suggests that the decreased ratings observed with repeated continuations in Experiment 1 (c.f., Figure 1) are unlikely to stem from the association

¹²<https://pickply.com/>

of the marker with the exhaustivity implicatures.

6 Discussion

We observed that some bigger models' results on the *nun* marker were partially coherent with human speakers' patterns. GPT-3—the largest model among the ones we assessed with log probabilities—showed sensitivity towards the non-cancelable implicature of the *nun* marker (Exp 1), although it did not exhibit sensitivity to the *lul* marker's cancelable exhaustivity implicature (Exp 1, 2). Polyglot-Ko-12B—second to the largest model—exhibited competency in utilizing the *nun* marker to evoke the exhaustivity implicature (Exp 2), but the model didn't show sensitivity towards the implicature's non-cancelability (Exp 1).

The current experiments evaluated LLMs' sensitivity towards discourse pragmatics, not the inferential pragmatics encoded with grammatical structures in a language. Most of the models we tested did not distinguish semantic contradictions from pragmatic infelicity encoded with linguistic structures, nor did they showcase the ability to alternate object markings to generate a particular discourse interpretation. GPT-3, the model trained as an instruction model, and ChatGPT, a model trained to follow conversational instructions, showed patterns closest to how humans associate discourse meanings with the two markers, albeit not identical. Both models demonstrated sensitivity to the discourse interpretation of the discourse marker *nun*, but not to the discourse interpretation of the canonically grammatical case marker *lul*.

In our results, larger-scale models—particularly those fine-tuned using RLHF—produced behavior that was more sensitive to the discourse meanings of morphological markers. This is consistent with past work suggesting that increases in model scale are correlated with improvements in performance (Kaplan et al., 2020). Of course, many factors were not controlled across the models we tested: the amount of training data, the architecture, whether the model was trained on multiple languages, and more. Future work would benefit from a finer-grained analysis of the corpus data that different models are trained on and examining the impact of the frequency of a given morphological marker with the model's ability to generate behavior sensitive to that marker's implicatures.

What does this result tell us about the general capabilities of distributional semantics in encod-

ing patterns of natural language? The particular challenge in the current experiments was that distributional semantics needed to encode dual meanings of morphological markers—grammatical object function and exhaustive interpretations established in the context. Distributional semantics, at least as operationalized in LLMs trained without human feedback, do not seem to capture how humans understand the dual functions of the markers. However, providing human feedback and scaling up the embedding space resulted in patterns closer to those in natural language. Thus, encoding dual meanings in multiple domains of language does not appear as an entirely impossible task for distributional semantics to handle.

7 Conclusion

The success of Large Language Models in various domains lends support to the hypothesis that much can be learned about the grammatical function and contextual meanings of words from their distributional patterns. In the current work, we examine whether distributional statistics are also sufficient to encode information about the discourse function of words or affixes in addition to their canonical meaning or function. Despite the proven competency of LLMs in grammatical domains, most models tested do not exhibit the human-like ability to use different structures in language to express nuanced meaning in the discourse context. Our study provides baseline assessments of what distributional semantics without any further fine-tuning could achieve.

Ethics Statement

All experiments with human participants followed the Institutional Review Board (IRB) guidelines at University of California San Diego. Data and codes are available at <https://github.com/hagyeongshin/lm-discourse/>.

Acknowledgements

The authors would like to thank Victor Ferreira for advice in human experiments, and the anonymous reviewers for their feedback. Any findings, opinions, and conclusions in this material are those of the authors.

References

- Judith Aissen. 2003. Differential Object Marking: Iconicity vs. Economy. *Natural Language & Linguistic Theory*, 21:435–483.
- Gemma Boleda. 2020. *Distributional Semantics and Linguistic Theory*. *Annual Review of Linguistics*, 6(1):213–234.
- Georg Bossong. 1991. *Differential Object Marking in Romance and Beyond*. In Dieter Wanner and Douglas A. Kibbee, editors, *New analyses in Romance linguistics*, pages 143–170. John Benjamins Publishing Company, Amsterdam.
- Daniel Büring. 2003. On D-Trees, Beans, And B-Accents. *Linguistics and Philosophy*, 26:511–545.
- Hye-Won Choi. 1996. *Optimizing structure in context: Scrambling and information structure*. Ph.D. thesis, Stanford University.
- Injoo Choi-Jonin. 2008. Particles and Propositions in Korean. In *Adpositions: Pragmatics, Semantic and Syntactic Perspectives*, number 74 in *Typological Studies in Language*, pages 133–170. John Benjamins Publishing Company, Amsterdam/Philadelphia.
- 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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.
- H. Paul Grice. 1989. *Studies in the Way of Words*. Harvard University Press, Cambridge, MA.
- Zellig S. Harris. 1954. *Distributional Structure*. *WORD*, 10(2-3):146–162.
- Laurence R Horn. 1981. Exhaustiveness and the Semantics of Clefts. In *Proceedings of the Eleventh Annual Meeting of the North Eastern Linguistic Society*, volume 11, pages 125–142.
- Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.
- Jennifer Hu and Roger Levy. 2023. Prompt-based methods may underestimate large language models’ linguistic generalizations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060. Association for Computational Linguistics.
- Paloma Jeretič, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are Natural Language Inference Models IMPPRESSive? Learning IMPLICature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. *Scaling Laws for Neural Language Models*. ArXiv:2001.08361 [cs, stat].
- Jieun Kim. 2018. *Deriving the contrastiveness of contrastive -nun in Korean*. *Linguistics and Philosophy*, 41(4):457–482.
- Song-Nim Kwon and Anne Zribi-Hertz. 2008. Differential Function Marking, Case, and Information Structure: Evidence from Korean. *Language*, 84(2):258–299.
- Knud Lambrecht. 1994. *Information Structure and Sentence Form: Topic, focus, and the mental representations of discourse referents*. Cambridge University Press.
- Chungmin Lee. 2003. Contrastive topic and/or contrastive focus. In *Japanese/Korean Linguistics*, volume 12, pages 352–364.
- Chungmin Lee. 2017. *Contrastiveness in Information Structure, Alternatives and Scalar Implicatures*. 91. Springer International Publishing, Cham, Switzerland.
- Hanjung Lee. 2006. Iconicity and Variation in the Choice of Object Forms in Korean. *Language Research*, 42(2):323–355.
- Hanjung Lee. 2007. *Case ellipsis at the grammar/pragmatics interface: A formal analysis from a typological perspective*. *Journal of Pragmatics*, 39(9):1465–1481.
- Alessandro Lenci. 2018. *Distributional Models of Word Meaning*. *Annual Review of Linguistics*, 4(1):151–171.
- Marco Marelli and Marco Baroni. 2015. *Affixation in semantic space: Modeling morpheme meanings with compositional distributional semantics*. *Psychological Review*, 122(3):485–515.
- Mats Rooth. 1992. *A theory of focus interpretation*. *Natural Language Semantics*, 1(1):75–116.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT Rediscovered the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? Probing for sentence structure in contextualized word representations.

Sean Trott and Benjamin Bergen. 2021. RAW-C: Relatedness of Ambiguous Words—in Context (A New Lexical Resource for English). 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 7077–7087, Online. Association for Computational Linguistics.

Sean Trott, Cameron Jones, Tyler Chang, James Michaelov, and Benjamin Bergen. 2023. *Do Large Language Models Know What Humans Know? Cognitive Science*, 47(7):e13309.

Robert Van Rooij and Katrin Schulz. 2004. *Exhaustive Interpretation of Complex Sentences. Journal of Logic, Language and Information*, 13(4):491–519.

A Items used in Experiment 1

An example of items provided to LLMs excluding ChatGPT.

소희는 미나와 유나가 메달과 트로피를 받을 수 있었다는 것을 알고 있습니다.
 소희: 유나는 뭘 받았어?
 미나: {메달을 받았어./메달은 받았어.} {트로피도 받았어./트로피는 못 받았어./트로피만 받았어.}

An example of items provided to ChatGPT, with the system message set as “반드시 1과 7사이의 숫자 중 하나로만 답해 주세요.”

소희는 미나와 유나가 메달과 트로피를 받을 수 있었다는 것을 알고 있습니다.
 소희: 유나는 뭘 받았어?
 미나: {메달을 받았어./메달은 받았어.} {트로피도 받았어./트로피는 못 받았어./트로피만 받았어.}
 위의 대화에서 미나가 첫 번째 답장에 이어 두 번째 답장을 말한 것이 얼마나 적절한지 1과 7 사이의 숫자로 답해 주세요. 1은 ‘전혀 적절하지 않다’는 것을 뜻하고 7은 ‘매우 적절하다’는 것을 뜻합니다.

An example of items provided to human participants.

소희, 미나, 유나가 함께 마라톤 경주에 참여하기로 했습니다. 마라톤 경주에 참여한 사람들은 메달과 트로피를 받을 수 있었습니다. 미나와 유나는 약속한 대로 함께 경주에 참여했습니다. 소희는 약속을 지키지 않았기 때문에 경주에서 누가 무엇을 받았는지 모르고 있습니다. 이후 소희와 미나는 다음 페이지에 나와 있는 문자 메시지를 주고 받았습니다.

아래에 소희와 미나가 마라톤 경주에 대해 이야기한 문자 메시지가 주어졌습니다.

소희: 유나는 뭘 받았어?
 미나: {메달을 받았어./메달은 받았어.} {트로피도 받았어./트로피는 못 받았어.}

위의 대화에서 미나가 첫 번째 답장에 이어 두 번째 답장을 말한 것이 얼마나 적절한다고 생각하십니까?

전혀 적절하지 않다 매우 적절하다
 1 2 3 4 5 6 7

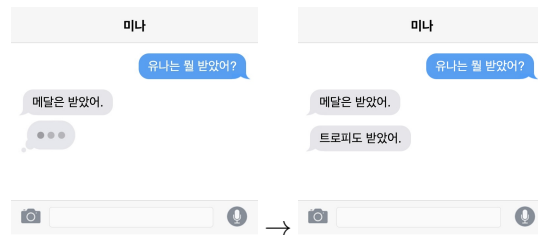


Figure 4: The question and answer portions in the human experiment items were presented in an interface resembling that of mobile text messages. Each message appeared in a 3-second interval within a short video clip.

B Raw ratings from Experiment 1

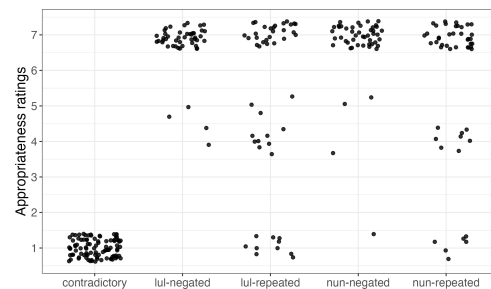


Figure 5: Raw ratings obtained from ChatGPT in Experiment 1 (1 = not appropriate at all, 7 = highly appropriate).

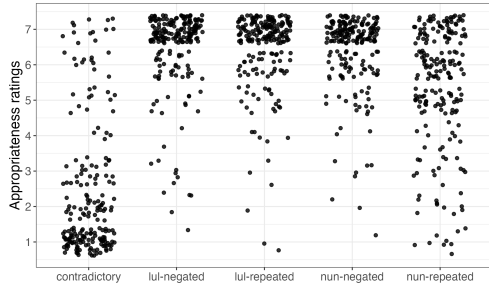


Figure 6: Raw ratings obtained from 34 human participants in Experiment 1 (1 = not appropriate at all, 7 = highly appropriate).

C Items used in Experiment 2

An example of items provided to LLMs, excluding ChatGPT.

{미나는 유나가 메달만 땀다고 답하려고 합니다./미나는 유나가 메달과 트로피를 둘 다 땀다고 답하려고 합니다.}
 소희: 유나는 뭘 받았어?
 미나: {메달을 받았어./메달 받았어./메달은 받았어./메달만 받았어./메달이랑 트로피를 둘 다 받았어.}

An example of items provided to ChatGPT, with the system message set as “일상적인 대화 상황을 생각하고 답장을 골라 주세요. 주어지는 답장들은 말해야 하는 모든 것을 나타내지 않을 수 있습니다. 이와 상관 없이 두 개의 답장 중에 더 적절한다고 생각하는 답장을 골라 주세요.”

{미나는 유나가 메달만 땀다고 답하려고 합니다./미나는 유나가 메달과 트로피를 둘 다 땀다고 답하려고 합니다.}
 소희: 유나는 뭘 받았어?
 미나: {메달을 받았어. 메달 받았어./메달은 받았어. 메달 받았어.}

An example of items provided to human participants.

소희, 미나, 유나가 함께 마라톤 경주에 참여하기로 했습니다. 마라톤 경주에는 메달과 트로피이 걸려 있었습니다. 미나와 유나는 약속한 대로 함께 경주에 참여했습니다. 미나는 유나가 메달을 땀고 트로피를 따지 못했다는 것을 알고 있습니다. 소희는 약속을 지키지 않았기 때문에 유나가 무엇을 땀는지 모르고 있습니다. 이후 소희는 미나에게

아래에 나와 있는 문자메시지를 보내왔습니다.

소희: 유나는 뭘 받았어?

미나는 유나가 메달만 땀다고 답하려고 합니다. 아래에 주어진 문장들 중 미나가 답장으로 보내기에 가장 적절한 것은 무엇입니까? 아래에 주어진 문장들은 미나가 말해야 하는 모든 것을 나타내지 않을 수 있습니다. 이와 상관 없이 미나가 보내기에 최선이라고 생각하는 답장을 선택해 주세요.

선택지1: ○ 메달을 받았어. ○ 메달 받았어.
 선택지2: ○ 메달은 받았어. ○ 메달 받았어.

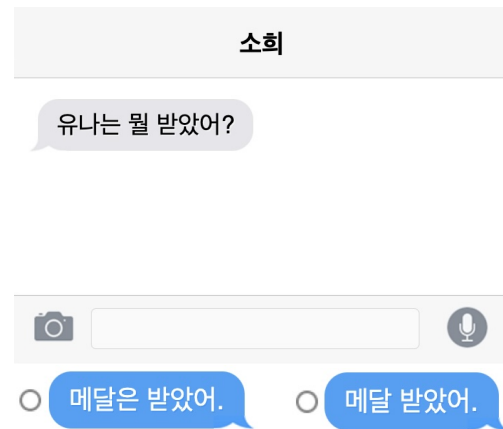


Figure 7: The question and response options in the human experiment items were presented in an interface resembling that of mobile text messages.

Computational Approaches for Integrating out Subjectivity in Cognate Synonym Selection

Luise Häuser

Computational Molecular Evolution group,
Heidelberg Institute for Theoretical
Studies, Heidelberg, Germany,
Institute for Theoretical Informatics,
Karlsruhe Institute of Technology, Karlsruhe, Germany
luise.haeuser@h-its.org

Gerhard Jäger

University of Tübingen
gerhard.jaeger@uni-tuebingen.de

Alexandros Stamatakis

Biodiversity Computing Group,
Institute of Computer Science, Foundation for Research and Technology - Hellas
Computational Molecular Evolution group,
Heidelberg Institute for Theoretical Studies, Heidelberg, Germany Institute for Theoretical Informatics,
Karlsruhe Institute of Technology, Karlsruhe, Germany
stamatak@ics.forth.gr

Abstract

Working with cognate data involves handling synonyms, that is, multiple words that describe the same concept in a language. In the early days of language phylogenetics it was recommended to select one synonym only. However, as we show here, binary character matrices, which are used as input for computational methods, do allow for representing the entire dataset including all synonyms. Here we address the question how one can and if one should include all synonyms or whether it is preferable to select synonyms a priori. To this end, we perform maximum likelihood tree inferences with the widely used RAxML-NG tool and show that it yields plausible trees when all synonyms are used as input. Furthermore, we show that a priori synonym selection can yield topologically substantially different trees and we therefore advise against doing so. To represent cognate data including all synonyms, we introduce two types of character matrices beyond the standard binary ones: probabilistic binary and probabilistic multi-valued character matrices. We further show that it is dataset-dependent for which character matrix type the inferred RAxML-NG tree is topologically closest to the gold standard. We also make available a Python interface for generating all of the above character matrix types for cognate data provided in CLDF format.

1 Introduction

Lexical data are frequently used as input to infer language trees via standard phylogenetic methods. Lexical data are typically cognate data that rely upon concept or meaning lists, such as the Swadesh List (Swadesh, 1955), for instance. When assembling data for a specific language, linguists typically attempt to identify a frequently used everyday word for each concept, which describes it as precisely as possible (Dunn, 2013). This data assembly process induces intrinsic difficulties, due to the inherent subjectivity in concept interpretation (List, 2018). Further, there often exist synonyms, that is, multiple words describe the same concept in a language and are used interchangeably by speakers. In German, for example, there are two words describing the concept "to kill": "töten" (related to the English word "dead") and "umbringen" (related to the English word "to bring"). The intricate nuances in meaning and usage are hard to determine and quantify (List, 2018). In the early days of language phylogenetics, Swadesh recommended to use the most common word only, "avoiding the complication of having to deal with a choice" (Swadesh, 1951). In "The ABC's of Lexicostatistics" (Gudschinsky, 1956), Gudschinsky advises to ensure objectivity by tossing a coin to decide which word to pick when there are several choices.

Computational phylogenetic methods have recently been applied to several cognate datasets. The inferences have mainly been conducted using Bayesian Inference (BI) methods (Kolipakam et al., 2018; Sagart et al., 2019; Heggarty, 2023), but a publication by (Jäger, 2018) shows that Maximum Likelihood (ML) based tree inference is useful as well, especially on extremely large language trees. For ML and BI approaches, that both heavily rely on the same type of phylogenetic likelihood calculations, cognate data are typically encoded via binary character matrices that represent the complete dataset including *all* synonyms. A question that has not been addressed to date is whether phylogenetic likelihood models as used in standard ML and BI tree inference can accommodate this data representation or whether it is preferable to choose synonyms via a labor-intensive, potentially error-prone, and subjective manual process beforehand.

Here, we focus on Maximum Likelihood (ML) tree inferences using the widely used RAxML-NG tool. Initially, working with empirical language data, we show that the topology of a tree inferred for a cognate dataset containing selected synonyms only can differ by up to 100% from the tree topology inferred for the corresponding complete dataset including all synonyms. Given these large potential discrepancies, we advise against manual selection. To alleviate this issue, we explore the potential benefits of using two types of alternative character matrices beyond the standard binary ones. The character matrices we propose can seamlessly be read as input by RAxML-NG while representing the complete dataset including all synonyms. We analyze the tree topologies resulting from ML inferences on all three character matrix types. We find that it depends on the respective dataset for which character matrix type the inferred tree best corresponds to the gold standard.

The remainder of this paper is structured as follows: First, we introduce our materials and methods. In particular, we formally define cognate data and describe the assembly process of the different character matrix types. Then, we evaluate how synonym selection affects the ML tree topologies inferred with RAxML-NG. Finally, we compare the introduced character matrix types. We consider the trees inferred with RAxML-NG and assess the matrix types based on how close the respective inferred trees are to the Glottolog gold standard.

2 Materials and Methods

2.1 Cognate Data

Each cognate dataset is based on a list of concepts. Collecting data for the languages under study results in an assignment of a set of words to each language-concept pair. From these data, we construct a matrix M containing the words' cognate classes. Cognate classes unite words that have been derived from a common ancestor (Dunn, 2013) (see Figure 1). We assume that the concept lists are reasonably assembled, that is, there exists at least one word for each concept in each language. When no word is given for a language-concept pair in the original data we interpret this as missing information.

If there are multiple synonym words describing a concept in a language, we say that the respective matrix cell is a *multi-state cell*. Otherwise, if there only is a single word describing a concept in a language, we call the respective matrix cell a *single-state cell*.

	big		big
English	big	E	big_1
English	great	E	big_2
German	groß	G	big_2
Dutch	groot	D	big_2
Norwegian	stor	N	big_3
Swedish	stor	S	big_3
	(a)		(b)

Figure 1: (a): Native cognate data (b): Corresponding matrix M with cognate classes (b)

2.2 Character Matrix Types

2.2.1 Binary Character Matrices

A cognate dataset can be represented by a binary character matrix A^b containing the symbols 0 and 1). Additionally, specific entries may be set to the undetermined character $-$, to represent missing information. Given $-$ at a certain column for a language, this language does not contribute anything to the respective per-column likelihood score in RAxML-NG. Hence, the missing entries do not affect the inference. However, the lack of information itself may impact the results (Roure et al., 2012).

We obtain A^b as the presence-absence-matrix corresponding to the matrix containing the cognate classes (see Figure 2 (a)). Each concept is therefore represented by as many columns as there are cognate classes, each corresponding to a specific cognate class. If there exists a word belonging to this cognate class in a certain language, the respective entry is set to 1, and to 0 otherwise. Thereby we assume that for each concept, there exists at least one word in every language. If there is no cognate class provided for a languages and a concept, this corresponds to missing information. We consequently set all columns corresponding to this concept to $-$.

2.2.2 Multi-valued Character Matrices

Some cognate datasets can also be represented by a multi-valued character matrix A^m containing multiple distinct symbols. In RAxML-NG, multi-valued character matrices are restricted to a maximum of 64 distinct symbols (Kozlov et al., 2019). In A^m , each concept is represented via a single data column only, but a different symbol is used for each cognate class. Multi-valued character matrices are thus restricted to represent a single cognate class for each language-concept pair. In order to construct A^m , the matrix M describing the respective cognate dataset must therefore contain single-state cells only. As this is generally not the case, we exclude multi-valued character matrices from further considerations.

	big_1	big_2	big_3		big
E	1	1	0	E	0, 1
G	0	1	0	G	1
D	0	1	0	D	1
N	0	0	1	N	2
S	0	0	1	S	2

(a) (b)

Figure 2: (a): Binary character matrix A^b (b): Multi-valued character matrix A^m , both corresponding to the cognate dataset in Figure 1. Note that A^m is invalid, because M (English, big) is a multi-state cell.

2.2.3 Probabilistic Character Matrices

The character matrices described so far are all deterministic, because we assume that a fixed symbol is observed at each data column for each language. In a probabilistic character matrix, we instead as-

sume that distinct symbols can occur with a certain probability, which is provided in the matrix. To represent missing data we explicitly set the probabilities for all symbols to 1.0 (Kozlov et al., 2019). This encoding does not provide any information, and hence, the missing entries do not contribute to the likelihood score.

We can represent a probabilistic character matrix in a file via the so-called CATG-Format that is supported by RAxML-NG (Kozlov et al., 2019). A tree inference based on a probabilistic character matrix differs from a standard inference with respect to the form of the conditional likelihood vectors at the tips. Usually, such a vector contains a single 1.0 entry for the observed discrete character while the remaining entries are all set to 0.0. In contrast, for a probabilistic character matrix, the conditional likelihood vectors are determined based on the provided probabilities (Kozlov et al., 2019) (see Figure 3).

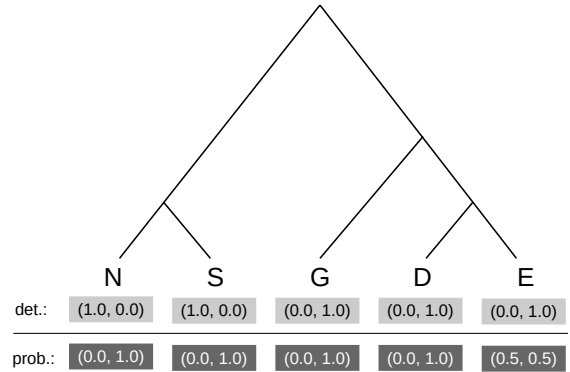


Figure 3: Tree with conditional likelihood tip vectors for the per-column likelihood of the column big_2. The light gray vectors refer to the inference on A^b (Figure 2), the dark gray ones to the inference on A^{b*} (Figure 4).

When interpreting cognate data in a probabilistic manner, we can represent the datasets via probabilistic character matrices. If k synonyms exist for a concept in a language, we can assume that each of them occurs with probability $\frac{1}{k}$. Based on these probabilities, we can subsequently assemble probabilistic binary and multi-valued character matrices. In the probabilistic binary character matrix A^{b*} , a concept is represented by as many columns as there are cognate classes, just as in the corresponding deterministic binary character matrix A^b . At a column corresponding to one of the k synonyms, we observe the symbol 1 with probability $\frac{1}{k}$ and the symbol 0 with probability $1 - \frac{1}{k}$ for the respective

language. In the multi-valued probabilistic character matrix A^{m*} , each cognate class related to a certain concept is encoded by a different symbol but the concept is represented by only one single column. At this column, we observe each of the symbols representing one of the k synonyms with probability $\frac{1}{k}$ (see Figure 4).

	big_1	big_2	big_3
E	(0.5, 0.5)	(0.5, 0.5)	(1.0, 0.0)
G	(1.0, 0.0)	(0.0, 1.0)	(1.0, 0.0)
D	(1.0, 0.0)	(0.0, 1.0)	(1.0, 0.0)
N	(1.0, 0.0)	(1.0, 0.0)	(0.0, 1.0)
S	(1.0, 0.0)	(1.0, 0.0)	(0.0, 1.0)

(a)

	big
E	(0.5, 0.5, 0.0)
G	(0.0, 1.0, 0.0)
D	(0.0, 1.0, 0.0)
N	(0.0, 0.0, 1.0)
S	(0.0, 0.0, 1.0)

(b)

Figure 4: (a): Probabilistic binary character matrix A^b (b): Probabilistic multi-valued character matrix A^m , both corresponding to the cognate dataset in Figure 1.

2.3 Comparing Trees

We measure topological dissimilarities between inferred phylogenetic trees using the *Robinson-Foulds (RF) distance* (Robinson and Foulds, 1981). This standard metric is based on non-trivial splits in trees. A split is a partitioning of the tree’s tips into two sets corresponding to the subtrees that arise when a branch of the tree is removed. A split is called non-trivial, if each set contains at least two tips. The absolute RF distance between two trees is defined as the number of non-trivial splits, which are unique to one of the two trees. In the following, we use the relative RF distance which we obtain by normalizing the absolute RF distance with $2(n-3)$, the total number of non-trivial splits in two strictly binary unrooted trees.

The inferred ML trees are strictly binary, but polytomies can occur in manually constructed reference trees (e.g., in Glottolog reference trees, see below). To compare an inferred ML tree to a reference tree, we therefore use the *generalized quartet (GQ) distance* (Pompei et al., 2011) instead. This metric

has the advantage that it yields a distance of 0 if there are no contradictions between the inferred tree and the reference tree, even if the reference tree does contain polytomies. To calculate the GQ distance, we extract all possible quartets of tips in the tree. For each quartet, we then determine the topology of the 4-tip subtree induced by the comprehensive tree. When comparing two trees, the GQ reflects the proportion of quartets that induce distinct topologies. Note that the RF distance and the GQ distance are distributed differently (Steel and Penny, 1993), which is illustrated by the following example: Let T be a fully balanced strictly binary tree with 16 leaves and let further T' be a tree obtained from T by swapping two leaves from neighboring subtrees (for details refer to Figure 11 in the supplement). The RF distance of T and T' is 0.15, while they exhibit a GQ distance of 0.03 only.

2.4 Maximum Likelihood Tree Inferences

In our experiments, we execute 20 independent ML tree searches on each character matrix under study. We use the default tree search configuration of RAxML-NG (10 searches starting from random trees and 10 searches starting from randomized stepwise addition order parsimony trees). For the inferences on both, the deterministic, and the probabilistic binary character matrices, we use the BIN+G model of binary character substitution. For the tree searches on probabilistic multi-valued character matrices we use the MK+G model which allows using up to 64 different characters but assumes equal substitution rates between all characters. Using BIN+G (MK+G resp.), we approximate the Γ model of rate heterogeneity via four discrete rates. Thus, each inference includes the ML estimation of the $\alpha \in [0, 100]$ shape parameter that determines the shape of the Γ distribution. The smaller the estimate of α , the higher the rate heterogeneity in the respective dataset will be (Yang, 1995). The command lines we used to execute the inferences are available on Github (<https://github.com/luisevonderwiese/synonyms>).

2.5 Quantifying Difficulty

To quantify the difficulty of a phylogenetic inference for a given dataset we use the difficulty score as predicted by Pythia (Haag et al., 2022). The tool internally uses a Random Forest Regressor (Ho, 1995) to predict this difficulty score based on attributes of the character matrices and on the

results of computationally substantially less expensive parsimony-based tree inferences (Farris, 1970). Because the parsimony approach can only be applied to deterministic character matrices, the difficulty score is also limited to this matrix type.

2.6 Data

For our analyses we use 44 cognate datasets. We retrieve the vast majority (39 datasets) from the cross-linguistic lexical database *Lexibank* (List et al., 2022). The five remaining datasets originate from the supplementary material provided for the book "Sequence Comparison in Historical Linguistics" (List, 2021). The above repositories comprise more datasets than we use here, as not all of them are suitable for our experiments. Since GQ distances can only be calculated on trees with strictly more than 4 tips, we exclude datasets comprising less than 5 languages. We also do not consider datasets with more than 400 languages due to the excessive tree inference times. In addition, these datasets exhibit an unfavorable number of concepts to number of languages ratio which yield them difficult to reliably infer. We therefore do not expect the respective tree inference results to be informative. We further exclude datasets with regard to the maximum number of different cognate classes that are occurring for the concepts. If only one cognate class is used for each concept, we do not consider the respective dataset, as the corresponding character matrices are not informative. We also exclude datasets comprising concepts with more than 64 distinct cognate classes, as no probabilistic multi-valued character matrix can be constructed in this case, because RAxML-NG is limited to using a maximum of 64 distinct symbols.

As gold standard, we use the manually constructed tree published by *Hammarström et al.* in the *Glottolog* database. This tree contains all 8205 languages listed in Glottolog. For each dataset, we obtain the respective gold standard tree by constraining the comprehensive tree to the languages contained therein. For a dataset to be suitable for our experiments, it must be possible to extract an informative reference tree. To this end, we exclude datasets, if the corresponding reference tree has a star topology because comparisons of binary topologies with a star topology do not yield meaningful topological distances.

In both data sources, the datasets are standardized as specified by the Cross-Linguistic Data

Format (CLDF) (Forkel et al., 2018). Our implementation for converting CLDF data into the character matrix types we describe here is available on Github (<https://github.com/luisevonderwiese/lingdata>).

3 Results

3.1 Effects of Synonym Selection

In the following, we aim to assess, whether ML tree inferences are feasible on binary character matrices representing cognate datasets with *all* synonyms or whether it is preferable to choose synonyms manually in advance. For this purpose, we investigate how it impacts the results of RAxML-NG based tree inferences when different combinations of synonyms are being selected. The experimental setup is illustrated in Figure 5

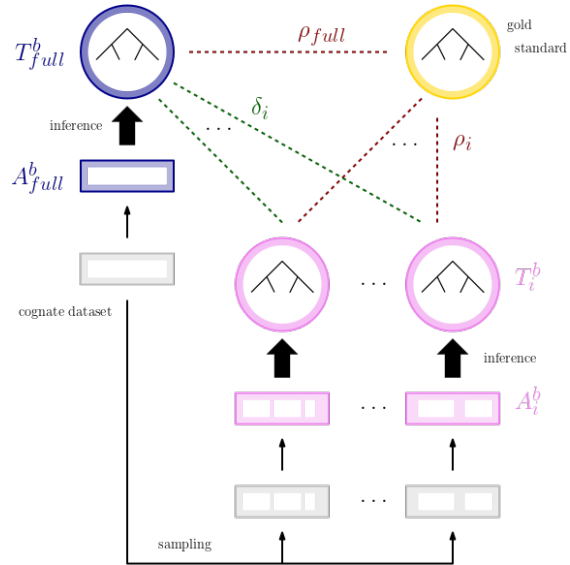


Figure 5: Experimental setup for assessing the effects of synonym selection: For each dataset, we create 100 selection samples and construct the corresponding (deterministic) binary character matrices A_i^b , $i \in 1, \dots, 100$. A_{full}^b represents the complete dataset including all synonyms. For each character matrix we consider the best scoring tree resulting from 20 independent tree searches with RAxML-NG, denoted by T_i , $i \in 1, \dots, 100$, T_{full} respectively. δ_i corresponds to the RF distance between T_i and T_{full} , ρ_i to the GQ distance between T_i and the gold standard tree. By ρ_{full} we denote the GQ distance between T_{full} and the gold standard.

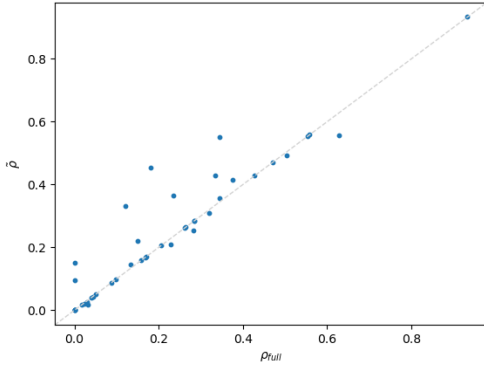


Figure 6: Each marker corresponds to a specific dataset under study. The x-axis indicates ρ_{full} , the GQ distance of the respective tree T_{full} to the gold standard. The y-axis indicates $\tilde{\rho}$, the median of the GQ distances of the trees T_i to the gold standard. For the majority of the datasets, the tree T_{full} is at least as close to the gold standard as the median of the trees T_i .

We generate selection samples from each examined dataset by selecting synonyms uniformly at random. Hence, if k synonyms exist for a concept in a language, we pick one of them with probability $\frac{1}{k}$. We create 100 such selection samples for each dataset under study. For some of the datasets, we performed the following experiments with 1000 samples, but this did not lead to substantial differences in the results. For each sample we construct a (deterministic) binary character matrix A_i^b , $i \in 1, \dots, 100$. Note that each of these matrices only contains information about the selected synonyms. Then, using RAxML-NG with the tree search options as outlined above we execute 20 independent ML tree searches on each of these character matrices as well as on the character matrix A_{full}^b representing the complete dataset including all synonyms. For each character matrix A_i^b , we consider the best-scoring ML tree T_i we inferred on it. Let further T_{full} be the best-scoring tree resulting for A_{full}^b . For each dataset, we use the corresponding gold standard tree from Glottolog as a reference. Let ρ_{full} be its GQ distance to T_{full} . For each tree T_i , we denote its GQ distance to the gold standard by ρ_i . Let further $\tilde{\rho}$ be the median of the GQ distances ρ_i . For ρ_{full} , we obtain an average distance of 0.22 over all 44 datasets while $\tilde{\rho}$ averages to 0.25. Thus, the two approaches appear to perform equally well at first sight, with T_{full} being only slightly closer to the reference, on average.

For a more detailed assessment, we compare ρ_{full} and $\tilde{\rho}$ for each dataset (see Figure 6). For 33 out of 44 examined datasets, we observe that $\rho_{\text{full}} \leq \tilde{\rho}$, that is T_{full} comes closer to the gold standard. In most of the cases, the inference on A_{full}^b thus performs better than the median inference on the sampled character matrices. For the datasets where $\rho_{\text{full}} > \tilde{\rho}$ applies, the difference never exceeds 0.07 and only for 5 datasets it exceeds 0.01. If the median tree T_i is closer to the gold standard, the differences are hence not substantial in most cases. These results speak in favor of performing inferences on the full dataset as the results tend to be slightly better than for the median randomized synonym selection. When using the mean GQ distance instead of the median, the observations are analogous. For details refer to Figure 12 in the supplement.

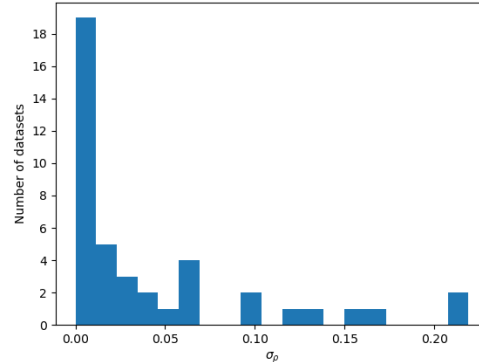


Figure 7: Distribution of σ_ρ , the standard deviation of the GQ distances of the trees T_i to the gold standard. The x-axis indicates σ_ρ , the y-axis the number of datasets such that respective standard deviation occurs. The observed standard deviations indicate that the inferred trees are substantially different depending on the subset of selected synonyms.

The advantages of using the full datasets are further emphasized when analyzing the stability of the tree inference under the described sampling procedure. For this purpose, we evaluate how much the trees T_i vary in terms of their deviation from the gold standard. For a fixed dataset, let σ_ρ be the standard deviation of the GQ distances ρ_i . The distribution of σ_ρ over all examined datasets is depicted in Figure 7. For 13 datasets, we observe $\sigma_\rho > 0.05$, for 2 of them, even $\sigma_\rho > 0.2$. Since the GQ distance is

distributed differently than the RF distance, the observed standard deviations indicate substantial differences. How close the inferred tree comes to the gold standard can therefore vary considerably depending on the subset of selected synonyms.

In an additional stability analysis, we examine, to which extent the trees T_i deviate from the respective tree T_{full} . For each tree T_i , we therefore determine δ_i as its RF distance to T_{full} . We consider $\bar{\delta} := (\sum_{i=1}^{100} \delta_i)/100$, indicated on the x-axis of Figure 8. The figure's y-axis gives the number of examined datasets for which the respective average distance results from the described experiment. $\bar{\delta}$ is close to 0 for some datasets but also ≥ 0.4 or even ≥ 0.8 for a considerable proportion of datasets. $\bar{\delta}$ can be interpreted as an indication for the stability of the inferred tree under sampling. The lower $\bar{\delta}$, the more stable the respective dataset is.

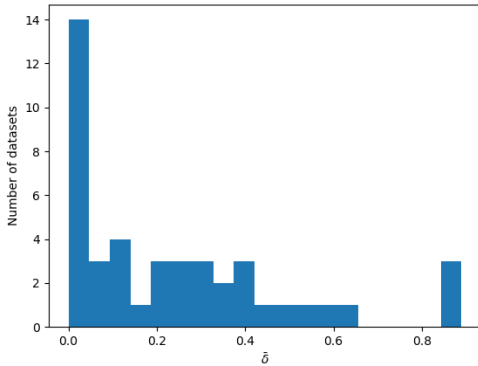


Figure 8: Distribution of $\bar{\delta} := (\sum_{i=1}^{100} \delta_i)/100$. The x-axis indicates $\bar{\delta}$, the y-axis the number of datasets such that the trees T_i yield the respective average RF distance to T_{full} . The lower $\bar{\delta}$, the more stable the respective dataset is under random synonym selection.

For each dataset we also determine the difficulty of the inference on the character matrix A_{full}^b . The obtained difficulty score is only slightly correlated with the stability as quantified by $\bar{\delta}$ (Pearson correlation 0.43, P-value 0.003). Further, we examine the proportion of multi-state cells for each cognate dataset, corresponding to the proportion of language-concept pairs such that there are words from more than one cognate class describing the concept in the language. This score is slightly correlated with $\bar{\delta}$ (Pearson correlation

0.45, P-value 0.002). The more multi-state cells exist, the more information are discarded during sampling and the more the sampled character matrix differs from the character matrix that is based on the full dataset, leading to the observed correlation.

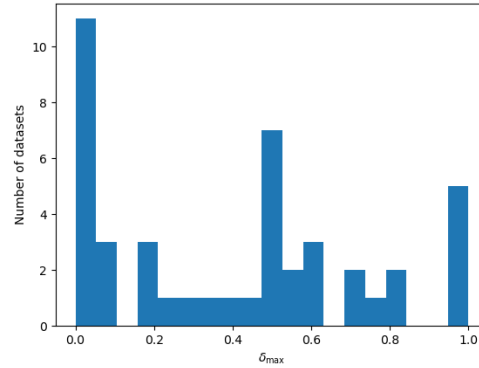


Figure 9: Distribution of $\delta_{\max} := \max(\delta_i : i \in 1, \dots, 100)$.

The x-axis indicates $\bar{\delta}$, the y-axis the number of datasets such that respective maximum value occurs among the RF distances of the trees T_i to T_{full} . The analysis illustrates that there exist datasets for which the worst case synonym choice leads to a tree which is entirely different (RF distance of 1.0) from the tree inferred on the dataset including all synonyms.

We finally consider $\delta_{\max} := \max(\delta_i : i \in 1, \dots, 100)$. The analysis of this score elucidates the detrimental effect of an unfavorable worst case synonym choice. Figure 9 depicts the respective distribution for the datasets under study. We observe that there is a considerable proportion of datasets with $\delta_{\max} \geq 0.5$. For 5 datasets, we even observe $\delta_{\max} = 1$. For these datasets there exists at least one character matrix containing a certain subset of synonyms such that the inferred tree admits an RF distance of 1.0 to the tree resulting from the full dataset.

Our observations illustrate that the synonym selection can induce entirely different RAxML-NG based ML tree topologies. The decision, which synonyms to consider thus substantially affects the results. Therefore, we strongly advise against manual synonym selection. Instead, we recommend to consider all known synonyms when inferring phylogenetic trees. Our analysis shows, that tree inference on the respective representation as a binary

character matrix leads to feasible results. Additionally, this circumvents the labor-intensive process of selecting synonyms manually.

3.2 Modeling Data with Synonyms

In the following, we compare the performance of ML tree inferences on three different kinds of character matrices representing cognate data. For each dataset under study, we consider its representation as a deterministic binary character matrix A^b , as a probabilistic binary character matrix A^{b^*} , and as a probabilistic multi-valued character matrix A^{m^*} . On each character matrix type we execute 20 independent ML tree searches with RAxML-NG as described above. The structure of the following experiment is illustrated in Figure 10.

We aim to assess how suitable the different character matrix types are for capturing the signal contained in the data during ML tree inference with RAxML-NG. To this end, we examine the resulting trees. For a fixed dataset, let T^b , T^{b^*} , T^{m^*} be the best-scoring tree inferred on the respective character matrix type. We compare these trees to the corresponding gold standard from Glottolog. We are interested in which character matrix type will induce the tree that is closest to it. We henceforth say that this is the character matrix type performing best.

The trees inferred on A^b yield an average GQ distance of 0.22 to the respective gold standard tree. For both probabilistic character matrix types we observe an average GQ distance of 0.23. At first glance, all character matrix types therefore appear to yield results of comparable quality. For 9 datasets, A^b performs best, for 9 datasets it is A^{b^*} , and for 11 datasets A^{m^*} . In 8 datasets, all character matrix types perform equally well. There are 7 datasets where two distinct character matrix types yield equally good results, but are better than the third. For the sake of simplicity, we exclude these datasets from our further analyses. There is no clear trend for one character matrix type always being preferable over the others. Conversely, based on our results, we cannot advise against the use of any type of character matrix.

Subsequently, we analyze the differences between the trees inferred on the different character matrix types, aiming to show that these differences are so substantial that it is indeed worthwhile to

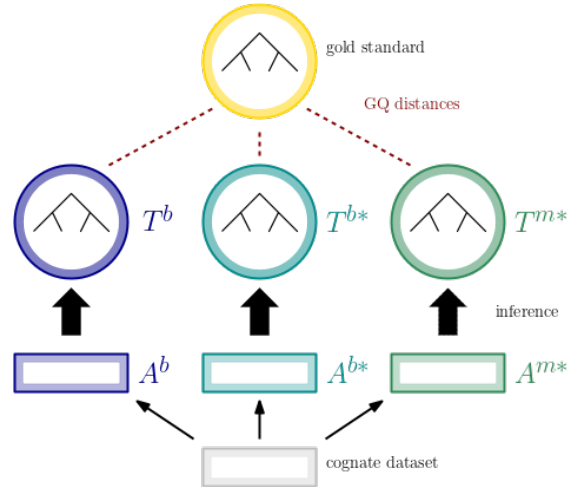


Figure 10: Experimental setup for comparing the performance of different character matrix types: For each dataset we construct a deterministic binary character matrix A^b , a probabilistic binary character matrix A^{b^*} , and a probabilistic multi-valued character matrix A^{m^*} . For each character matrix type, we consider the best tree scoring tree resulting from 20 independent tree searches with RAxML-NG, denoted by T^b , T^{b^*} , T^{m^*} respectively. We compute the GQ distances of these trees to the gold standard from Glottolog.

perform inferences on all three character matrix types. For this purpose we first consider the datasets, for which A^b performs best. Over these datasets, T^b , which comes closest to the gold standard tree, yields an average RF distance of 0.60 to $T_d^{b^*}$ and of 0.61 to $T_d^{m^*}$. This indicates that the trees resulting from the inferences on the various character matrix types can differ considerably. We observe an analogous behavior for the datasets where A^{b^*} or A^{m^*} perform best (see Table 1 in the supplement). We conclude that the differences between the inferred trees can become so large such that all three character matrix types always need to be considered.

In the following, we attempt to identify properties, that for a given dataset, might be able to predict, which character matrix type performs best. First, we consider the α shape parameter that determines the shape of the Γ distribution under the BIN+G (MK+G resp.) model. For 9 out of the 11 datasets with A^{m^*} performing best, the ML estimates of α are below 20, indicating a moderate to high degree of rate heterogeneity. This is also the case for 8 out of the 9 datasets for which A^{b^*} performs best. However, $\alpha < 20$ is only observed

for 5 out of the 9 datasets for which the binary character matrix yields the best performance and only for 4 out of the 8 datasets for which all character matrix types perform equally well. We therefore observe a tendency for probabilistic modeling to be advantageous for datasets with high rate heterogeneity. These datasets may exhibit a larger variance with respect to the number of cognate classes per concept, which can possibly be better accommodated via a probabilistic character matrix type.

We further investigate whether the difficulty of the ML inference on a certain dataset is related to character matrix performance. In the following, the difficulty reported for a dataset is the difficulty obtained for A^b , as difficulty scores can only be computed for deterministic character matrices. The datasets for which all character matrix types perform equally well exhibit a comparably low average difficulty of 0.17. A low difficulty score indicates a strong phylogenetic signal in the data. This strong signal can be captured during the ML tree inference, regardless of the type of character matrix used to represent it. The datasets with A^b performing best exhibit an average difficulty of 0.45. For the datasets for which A^{b*} performs best, the average difficulty is 0.29, and for those for which A^{m*} yields the best performance, it amounts to 0.18. While the probabilistic character matrix types are hence advantageous for data with a clear signal, the deterministic representation is better suited to capture the signal in datasets where this is more difficult.

4 Conclusion

We demonstrated, that the selection of synonyms in cognate datasets *can* induce substantially different tree topologies when performing ML inferences with RAxML-NG. It is thus preferable to perform tree inferences on the full dataset with all synonyms included. This also circumvents the labor-intensive process of manual synonym selection. The datasets can be encoded as (deterministic) binary character matrices. In addition, we introduced probabilistic binary and probabilistic multi-valued character matrices as alternative representations. We showed that it is dataset-dependent, for which character matrix type the inferred tree is closest to the gold standard. We were able to identify the rate heterogeneity and the difficulty score as properties

that may indicate which character matrix type is best suited for a given dataset. Note that unfortunately, the number of available cognate datasets is too low in order to train any machine-learning based predictors. We therefore recommend performing inferences on all three character matrix types when analyzing cognate datasets. We provide a Python interface on Github (<https://github.com/luissevonderwiese/lingdata>) that can be used to create all of the above character matrices for any cognate dataset provided in CLDF format.

5 Future Work

Our work leads to several novel questions that need to be addressed. When constructing probabilistic character matrices, data from corpus studies could be taken into account instead of assuming a uniform probability distribution of synonym occurrence, albeit this information could be challenging to obtain for small languages and dialects. Future work should further strive to develop an alternative model of evolution taking idiosyncrasies of language data into account. Another open question is how the quality of the inference methods can be assessed without referring to the gold standard. This requires the development of language-specific data simulation tools, taking into account the challenges that have been described with respect to simulating realistic DNA data (Troost et al., 2023).

Acknowledgement

Luise Häuser and Alexandros Stamatakis are financially supported by the Klaus Tschira Foundation, and by the European Union (EU) under Grant Agreement No 101087081 (CompBiodiv-GR). Gerhard Jäger is supported by the ERC-AdG 834050 (CrossLingference) and the DFG-FOR 2234 *Words, Bones, Genes, Tools*.

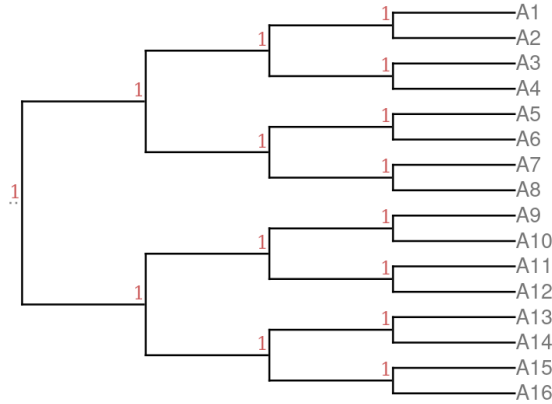


**Funded by
the European Union**

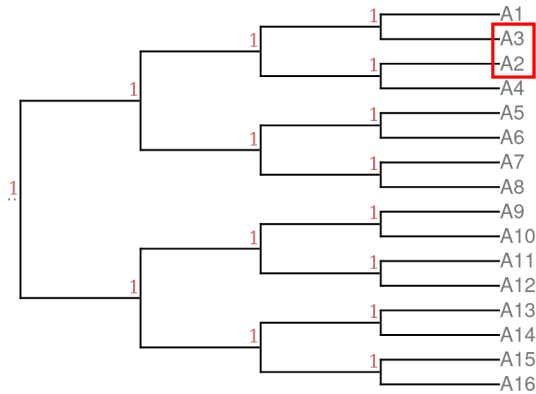
References

- Michael Dunn. 2013. Language phylogenies.
- James S. Farris. 1970. Methods for computing wagner trees. *Systematic Zoology*, 19(1):83–92.
- Robert Forkel, Johann-Mattis List, Simon Greenhill, Christoph Rzymiski, Sebastian Bank, Michael Cysouw, Harald Hammarström, Martin Haspelmath, Gereon Kaiping, and Russell Gray. 2018. [Cross-linguistic data formats, advancing data sharing and re-use in comparative linguistics](#). *Scientific Data*, 5:180205.
- Sarah C. Gudschinsky. 1956. The abc’s of lexicostatistics (glottochronology). *WORD*, 12:175–210.
- Julia Haag, Dimitri Höhler, Ben Bettisworth, and Alexandros Stamatakis. 2022. [From easy to hopeless - predicting the difficulty of phylogenetic analyses](#). *bioRxiv*.
- Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2022. [Glottolog 4.7](#).
- Paul et al. Heggarty. 2023. [Language trees with sampled ancestors support a hybrid model for the origin of indo-european languages](#). *Science*, 381(6656).
- Tin Kam Ho. 1995. [Random decision forests](#). In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.
- Gerhard Jäger. 2018. [Global-scale phylogenetic linguistic inference from lexical resources](#). *Scientific Data*, 5.
- Vishnupriya Kolipakam, Fiona M. Jordan, Michael Dunn, Simon J. Greenhill, Remco Bouckaert, Russell D. Gray, and Annemarie Verkerk. 2018. [A Bayesian phylogenetic study of the Dravidian language family](#). *Royal Society Open Science*, 5(171504):1–17.
- Alexey M Kozlov, Diego Darriba, Tomáš Flouri, Benoit Morel, and Alexandros Stamatakis. 2019. [RAXML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference](#). *Bioinformatics*, 35(21):4453–4455.
- Johann-Mattis List, Robert Forkel, Simon Greenhill, Christoph Rzymiski, Johannes Englisch, and Russell Gray. 2022. [Lexibank, a public repository of standardized wordlists with computed phonological and lexical features](#). *Scientific Data*, 9:316.
- Mattis List. 2018. [Tossing coins: linguistic phylogenies and extensive synonymy](#).
- Mattis List. 2021. *Sequence Comparison in Historical Linguistics*. düsseldorf university press, Berlin, Boston.
- Simone Pompei, Vittorio Loreto, and Francesca Tria. 2011. [On the accuracy of language trees](#). *PloS one*, 6:e20109.
- D.F. Robinson and L.R. Foulds. 1981. [Comparison of phylogenetic trees](#). *Mathematical Biosciences*, 53(1):131–147.
- Béatrice Roure, Denis Baurain, and Hervé Philippe. 2012. [Impact of Missing Data on Phylogenies Inferred from Empirical Phylogenomic Data Sets](#). *Molecular Biology and Evolution*, 30(1):197–214.
- Laurent Sagart, Guillaume Jacques, Yunfan Lai, Robin Ryder, Valentin Thouzeau, Simon J. Greenhill, and Johann-Mattis List. 2019. [Dated language phylogenies shed light on the ancestry of sino-tibetan](#). *Proceedings of the National Academy of Science of the United States of America*, 116:10317–10322.
- Mike Steel and David Penny. 1993. [Distributions of tree comparison metrics—some new results](#). *Systematic Biology - SYST BIOL*, 42:126–141.
- Morris Swadesh. 1951. [Diffusional cumulation and archaic residue as historical explanations](#). *Southwestern Journal of Anthropology*, 7(1):1–21.
- Morris Swadesh. 1955. [Towards greater accuracy in lexicostatistic dating](#). *International Journal of American Linguistics*, 21(2):121–137.
- Johanna Trost, Julia Haag, Dimitri Höhler, Laurent Jacob, Alexandros Stamatakis, and Bastien Boussau. 2023. [Simulations of Sequence Evolution: How \(Un\)realistic They Are and Why](#). *Molecular Biology and Evolution*, 41(1):msad277.
- Z Yang. 1995. [A space-time process model for the evolution of dna sequences](#). *Genetics*, 139(2):993–1005.

A Supplementary Figures and Tables



(a) Tree T , fully balanced strictly binary tree with 16 leaves



(b) Tree T' , differences to T are highlighted in red

Figure 11: The trees T and T' exhibit an RF distance of 0.15 but a GQ distance of 0.03, which illustrates the different distributions of the metrics.

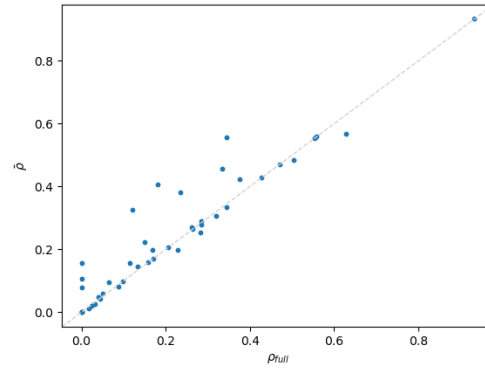


Figure 12: Each marker corresponds to a specific dataset under study. The x-axis indicates ρ_{full} , the GQ distance of the respective tree T_{full} to the gold standard. The y-axis indicates $\bar{\rho}$, the average GQ distance of the trees T_i to the gold standard. Most markers are located on the identity (represented by a dashed line) or above, that is the tree T_{full} is at least as close to the gold standard as the average tree T_i for the majority of the datasets.

Mean RF(\cdot, \cdot)	T^b	T^{b*}	T^{m*}
Datasets with A^b performing best:			
T^b	0.00	0.60	0.61
Datasets with A^{b*} performing best:			
T^{b*}	0.25	0.00	0.36
Datasets with A^{m*} performing best:			
T^{m*}	0.43	0.40	0.00

Table 1: Mean RF distances among the trees resulting from the inferences on the different character matrix types. Datasets are grouped according to the character matrix type performing best. The differences between the trees are so substantial that it is indeed worthwhile to perform inferences on all three character matrix types.

Stranger than Paradigms

Word Embedding Benchmarks Don't Align With Morphology

Timothee Mickus

University of Helsinki
timothee.mickus@helsinki.fi

Maria Copot

Ohio State University
copot.1@osu.edu

Abstract

Word embeddings have proven a boon in NLP in general, and computational approaches to morphology in particular. However, methods to assess the quality of a word embedding model only tangentially target morphological knowledge, which may lead to suboptimal model selection and biased conclusions in research that employs word embeddings to investigate morphology. In this paper, we empirically test this hypothesis by exhaustively evaluating 1,200 French models with varying hyperparameters on 14 different tasks. Models that perform well on morphology tasks tend to differ from those which succeed on more traditional benchmarks. An especially critical hyperparameter appears to be the negative sampling distribution smoothing exponent: Our study suggest that the common practice of setting it to 0.75 is not appropriate: its optimal value depends on the type of linguistic knowledge being tested.

1 Introduction

Word embeddings have changed the NLP landscape by introducing a data-driven approach to meaning. They have found widespread application in NLP, computational semantics, and more recently, morphology (e.g. Zeller et al., 2014; Bonami and Guzmán Naranjo, 2023).

While architectures specifically intended to capture morphology exist (Cao and Rei, 2016; Cotterell et al., 2016; Cotterell and Schütze, 2015), embeddings with these properties are generally not employed because not available off the shelf pre-trained on the languages of interest to the morphologist. A notable exception is fastText (Bojanowski et al., 2017), an architecture specifically tailored to factor in spelling information which has been tested on a diverse and wide collection of languages (Grave et al., 2018). Despite claims that this architecture is suited to model morphology due to its attention to subword information, this has rarely been

properly tested on morphological benchmarking. Additionally, this type of embedding is explicitly avoided by researchers who do not wish to smuggle in the assumption that the units of morphology are primarily based on formal contrasts, rather than on more abstract contrasts of meaning (as argued by e.g. Štekauer, 2014).

The adoption of word embeddings in morphological research has therefore largely targeted general purpose embeddings, with architectures that are not optimised for capturing morphological structure. However, the evaluation of these models mostly relies on tasks that were not built with morphology in mind. Common NLP benchmarks used by models for morphological purposes generally target semantics: To take a concrete example, Lenci et al. (2022) provide an exhaustive evaluation of distributional semantics models on a wide array of tasks. They study a spate of benchmarks targeting target semantics, such as synonymy detection, analogy solving, sentiment analysis and natural language inference; but only two of their tasks involve morphology: the analogy task (whose methodological and ethical limitations are well documented, e.g., Linzen, 2016; Bolukbasi et al., 2016); and POS-tagging (where some morphological knowledge may be of use, although it is not explicitly required). This trend may be in part ascribed to the Anglo-centric approach of most NLP research: English is a language with relatively scarce inflectional morphology, which therefore has received comparatively little interest from morphologists interested in the subject.

The tension between the increasingly widespread use of general-purpose word embeddings in morphology and their evaluation on non-morphological benchmarks begs the engineering question of how to adapt the knowledge the community has developed for English to other languages, in a way that encompasses morphological applications in addition to semantic

ones. In the present paper, we investigate whether a discrepancy exists between NLP evaluation methodologies and morphology applications of word embeddings. We define eight tasks, probing for both inflection and derivation, evaluating both the geometry of the vector space and its usability in downstream scenarios, and exhaustively compare the behavior of 1200 continuous bag-of-words negative sample embedding models (Mikolov et al., 2013, “CBOW-NS”) on traditional NLP semantic benchmarks as well as our proposed morphology tasks. We find that optimal hyperparameter settings are task-specific, and that there is a tradeoff between performance on tasks targeting different kinds of linguistic knowledge. We also stress the importance of the negative sampling distribution smoothing exponent hyperparameter, which we find to have a crucial role in our experiments despite its lack of notoriety.

2 Related works

Systematic studies of word embeddings. Works attempting to exhaustively evaluate word embeddings abound. These studies often delineate their area of focus to a specific architecture, language or hyperparametrization. For instance, Vulić et al. (2020) extensively study BERT models across six languages and five tasks. On the other hand, Lenci et al. (2022) provide an exhaustive overview of multiple English embeddings, across a diverse array of tasks and hyperparameters. Ulčar et al. (2020) and Grave et al. (2018) both limit their studies to fastText embeddings and the analogy task, but cover 9 and 10 languages respectively. Lastly, especially relevant to our present inquiry is the work of Köhn (2015), who focuses on the (morpho-)syntactic features captured in a diverse array of embedding architectures for Basque, English, French, German, Hungarian, Polish, and Swedish.

Architectures that capture morphology. A significant focus of interest concerns the development of embedding architectures designed to specifically capture some aspects of morphology. Chief of these is the fastText model of Bojanowski et al. (2017), which supplements the skip-gram model of Mikolov et al. (2013) with subword information. Cao and Rei (2016) propose an unsupervised character-level method that ranks each segment by its context-predictive power to capture information about morphological boundaries as well as morphological features. Cotterell et al. (2016) introduce

a semisupervised architecture trained on a combination of raw and morphologically annotated text, which creates embedding spaces where morphologically similar words cluster together. Cotterell and Schütze (2015) present a latent-variable Gaussian graphical model trained on an embedding set and a lexical resource to smooth an existing set of word embeddings in a way that encourages the encoding of morphology. With the exception of Bojanowski et al.’s (2017) fasttext, these models have not yet reached widespread adoption among morphologists—in part due to their restricted typological coverage, as exemplified by the challenges non-concatenative morphology poses for subword-centric approaches (e.g., Amrhein and Sennrich, 2021).

Word embeddings and morphology. Word embeddings are a somewhat recent adoption in the study of morphology. A short survey of the literature outlines three main use-cases for embeddings.

The first case involves using the features of trained embeddings as input to prediction tools, with the aim to create resources or investigate the morphological system. One such instance is Zeller et al. (2014) employ embeddings to validate the construction of a derivational lexicon. Straka and Straková (2017) details the use of embeddings as input features for tasks where morphology is relevant, such as lemmatization or tokenization. Bafna and Žabokrtský (2022) study how subword embeddings can be used for cross-lingual transfer between morphologically similar, diachronically related languages. Another related approach is that of Marelli and Baroni (2015), who propose to learn linear maps to model affixation.

Related but distinct from this approach, a second set of works use embeddings as tools for gathering quantitative evidence about morphology. A variety of topics have been covered: Lapesa et al. (2018) quantitatively assess the difference between eventive and non-eventive *-ment* formations in French; Guzmán Naranjo and Bonami (2021) rely on embeddings to discuss overabundance; Varvara et al. (2021) addresses the question of semantic transparency; Bonami and Guzmán Naranjo (2023) derive quantitative evidence in favor of a paradigmatic conception of derivation from embeddings.

The third case is the use of embeddings for the purposes of defining a morphologically coherent group of items by the properties of the position they occupy in the geometrical space—the analysis of

neighborhoods thus constructed may be qualitative (e.g. Wauquier, 2020) or quantitative (e.g. Huyghe and Wauquier, 2020). Varvara (2017) uses distributional representations to quantitatively evaluate neighborhood contents, and how they differ for event nominalizations and verbal nouns, A related trend of research involves performing operations on embeddings directly to derive quantifiable data—e.g., to study the difference between inflection and derivation (Bonami and Paperno, 2018; Rosa and Žabokrtský, 2019) or the status of specific morphological processes (Mickus et al., 2019).

3 Methodology

We set out to answer the question of whether it is in fact problematic to evaluate models we use for morphology on tasks which chiefly target lexical semantics. We do so by evaluating the performance of the same model on a diverse range of tasks targeting different kinds of linguistic knowledge. Because of its rich morphology and availability of resources documenting morphological relations, we elect to work on French. We wish to make as few assumptions as possible about whether we expect any systematic differences in performance between tasks and about what they might be caused by should they manifest—we adopt a grid-search approach and evaluate models trained with an exhaustive combination of values for a wide range of hyperparameters.

Models We train Continuous Bag-Of-Word negative sample models (Mikolov et al., 2013, CBOW-NS). Models are implemented with gensim (Řehůřek and Sojka, 2010), trained on a 300M French sentences subset of Oscar (Ortiz Suárez et al., 2019) We include a presentation of the word2vec algorithm and a few remarks on the linguistic significance of its hyperparameters in Appendix A.

Models defined with varying hyperparameters:

- (i) window size $w \in \{5, 10, 15, 20, 25\}$;
- (ii) number of negative examples per positive example $N \in \{5, 10, 15, 20, 25\}$;
- (iii) number of epochs $e \in \{1, 3, 5\}$;
- (iv) negative sampling distribution exponent $\alpha \in \{-1.4, -1.0, -0.6, -0.2, 0.2, 0.6, 1.0, 1.4\}$;
- (v) dynamic uniform sampling of window size $s \in \{\text{True}, \text{False}\}$.

All models have a dimension of $d = 50$, which we do not modify so as to avoid spurious concen-

tration effects in higher-dimensional spaces.¹ All combination of hyperparameters are tested, for a total of 1200 different models. As hyperparameters (i), (ii) and (iii) are frequently encountered in the literature, we refer the reader to the original paper by Mikolov et al. (2013) as well as to Appendix A.2 for details.

The negative sampling smoothing hyperparameter α in (iv) is not frequently tuned, but Caselles-Dupré et al. (2018) suggest it might have application-specific relevance. It is used to define the probability distribution q under which negative examples are randomly sampled:

$$q(w) \propto p(w)^\alpha$$

where $p(w)$ is the relative frequency of each word in the training corpus. Mikolov et al. (2013) note that α allows one to mix unigram and uniform distributions over vocabulary items: Setting a value closer to 0 allows one to sample more from the tail of the vocabulary’s frequency distribution. More precisely, remark that $\alpha = 0$ entails sampling negative examples uniformly over the entire vocabulary sorted by frequency; $\alpha = 1$ matches the unigram frequency distribution in corpus; $\alpha > 1$ over-emphasizes frequent words, and $\alpha < 0$ over-emphasizes infrequent ones. The relative dearth of studies on the effects of α on CBOW-NS representations to this day motivates us to be particularly thorough when testing this hyperparameter.

The dynamic uniform sampling s in (v) is a gensim-specific re-implementation of the distance-based weighting of context words. It consists in randomly sampling, for each training example, an effective window size \hat{w} uniformly between 1 and the maximum window size parameter allowed by the w hyperparameter, or more formally $\hat{w} \sim U(1, w)$. In practice, this entails that context words that are $k \leq w$ tokens apart from the target word are discarded in k/w of the training instances. Therefore, context words that are closer to the target word are more likely to be taken into account for prediction.

Common NLP benchmarks. All models are tested on the SimLex-999 French translation by Barzegar et al. (2018), the FEEL lexicon of Abdaoui et al. (2017), the automatic translation to French of the Google Analogy Test Set (GATS) provided by Grave et al. (2018), and a POS-tagging

¹This low value of d mitigates the computational costs of running exhaustive experiments. For the same reason, models varying across epoch e only correspond to different checkpoints of the same training procedure.

task. For GATS results, we separately keep track of the accuracy on three groups of analogical relations: semantic, derivational and inflectional;² groups can be found in Appendix B.1. The POS-tagging data was selected from the French section of OMW (Bond and Paik, 2012), by selecting, for each lemma, all POS-tags it could correspond to.

Results for PoS-tagging and FEEL correspond to macro-F1 scores of multi-layer perceptrons³ trained to predict the labels provided in the resource as binary vectors. Performance on SimLex-999 is evaluated as the Spearman correlation between human rating and cosine similarity. GATS performance corresponds to a 3CosAdd on a vocabulary restricted to the 300k most frequent words.

Inflectional tasks. To test a model’s performance on inflectional morphology specifically, we set up four different tasks. Given that the community uses embeddings both as features (predictors to plug into another models, e.g., Straka and Straková, 2017) and as representations for manual exploration (e.g. Wauquier, 2020), we consider both classifier-based tasks and geometric evaluations. A second orthogonal distinction is whether these probing tasks involve one word form or multiple word forms at once: this, in essence, captures distinct approaches to morphology, depending on whether they focus on individual words or relationships between them. Data for all four of these tasks consisted of verbal paradigms taken from the GLàFF (Hathout et al., 2014), a large inflected lexicon of French. The data set used focused on words without homographs, and cells that are in current use in the French language. Only words that had more than 50 occurrences in our Oscar sample were included in the testing; cf. also Appendix B.2.

The first task involves a classifier over singular items: In our single cell prediction (SCP) task, we classify input verb forms depending on which paradigm cell they correspond to. The second task, a paired cells prediction (PCP) task, consists in predicting whether two input verb forms correspond to the same paradigm cell. We compare models on these two predictions tasks using macro-F1. In our third task, a single cell clustering (SCC) task, we assess with silhouette scores whether the embeddings of forms cluster according to their cell. Lastly, in

²The latter two are often grouped in a “syntactic” category; here we follow the taxonomy of Gladkova et al. (2016).

³One 25D layer with ReLU activation, optimized with Adam (lr. of 0.001, $\beta = (0.9, 0.999)$) for up to 10,000 iterations, implemented in `scikit-learn` (Pedregosa et al., 2011).

our fourth task, a paired cell clustering (PCC) task, we report the silhouette score obtained by clustering pairs of forms depending on which relation they instantiate. For this last PCC task, we define pairs of verb forms as matrices of shape $[2 \times d]$, distance between two pairs P_A and P_B is then computed using the Froebenius norm $\|P_A - P_B\|_F$.

Derivational tasks. To evaluate how accurately models capture derivational morphology, we set up four tasks. Data for these tasks was taken from Namer et al. (2023), a database of French derivational relationships. They feature a variety of relationships between different parts of speech, reported in Appendix B.2. Semantics labels are attributed by grouping formal exponents in the resource following the clustering proposed by Guzman Naranjo and Bonami (2023). Only words that had more than 50 occurrences in our Oscar sample were kept; cf. Appendix B.2 for details.

Following the same logic as for our inflectional task, we consider two prediction tasks and two classification tasks. Independently from this, we also note that there is ongoing discussion in the theoretical morphology community about whether derivational relations should be defined by means of formal or semantic regularity (Štekauer, 2014). We therefore decide to consider as labels either the formal exponent of the derived form (e.g. *-ité*), or the semantics associated with it (e.g., adjective-to-property-noun conversion). The two predictions tasks are set up as simple logistic regression classifiers that predict the derivational cell (defined based on semantics vs formal exponents for DerPS and DerPF respectively); we report the corresponding macro-F1 scores. The two clustering tasks reemploy the same protocol as the PCC inflectional task: we construct $[2 \times d]$ matrices for each derivationally related pair in our dataset, and compute the silhouette score for clustering them along their exponents in the DerCF task or the semantics of the process in the DerCS task.

4 Results

Given the high number of models and tasks, we first study how and to what extent specific hyperparameters shape performance; we defer an overview of actual performances to Appendix C.2 to focus primarily on global trends. To attribute the observed variance across scores to specific factors, we apply gradient boosting trees (Friedman, 2001) to the set of all (task-standardized) scores, using as predic-

α	task	e	N	w	s
0.69	0.23	0.18	0.08	0.08	0

Table 1: mean absolute SHAP value from boosting trees predicting performance.

Task	top 1	top 10		top 100	
		mean	median	mean	median
simlex	0.20	0.20	0.20	0.20	0.20
FEEL	0.20	0.24	0.20	0.36	0.20
GATS/sem	0.20	0.20	0.20	0.09	0.20
GATS/D	0.20	0.20	0.20	0.24	0.20
DerCF	0.60	0.76	0.60	0.70	0.60
DerCS	0.20	0.24	0.20	0.35	0.20
DerPF	0.60	0.64	0.60	0.73	0.60
DerPS	0.60	0.80	0.80	0.80	1.00
POS	0.60	0.88	1.00	0.84	1.00
GATS/I	0.20	0.28	0.20	0.36	0.20
SCP	1.40	1.40	1.40	1.20	1.40
PCP	1.40	1.40	1.40	1.32	1.40
SCC	1.00	1.24	1.40	1.06	1.00
PCC	1.00	0.88	1.00	0.86	1.00

Table 2: The mean and median α of the top 1, top 10 and top 100 performing models for each task.

tors the hyperparameters as well as the task, before computing SHAP values (Lundberg and Lee, 2017). Corresponding results can be seen in Table 1: The model had a residual mean standard error (RMSE) of 0.17 on the test set (one third of the data). Remarkably, the most important predictor was found to be the negative sampling distribution smoothing exponent α , with a mean absolute SHAP value of 0.69. Across most tasks, we find that values of α tend to produce natural clusters of model scores (see Figure 2 in Appendix C).

A summary of the distribution of performance for α values by task is reported in Table 2. We observe that common NLP benchmarks (FEEL, simlex, as well as all categories of analogies in GATS), appear to benefit from an α value of 0.2, while the tasks we devised to target inflectional morphology fare best with $\alpha \geq 1$. Derivational tasks lie somewhere in between: in DerCS, where processes are grouped by their semantics pattern close to semantic tasks, the optimal α is slightly higher than 0.2; in the three other tasks, optimal α values range from 0.6 to 0.8. POS appears to perform best with values in between those of inflection and derivation, with α slightly lower than 1. Data from GATS does not pattern as expected given the analogical relation type.

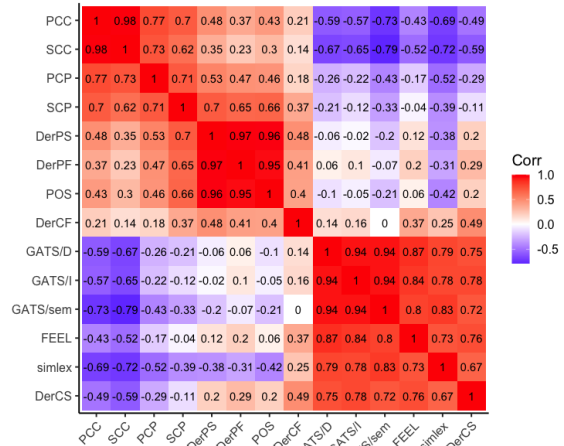


Figure 1: Spearman correlation for performance of models with $\alpha > 0$ on the different tasks.

We can further observe that values of $\alpha < 0$ tend to yield lower scores: Mann-Whitney U-tests indicate that with the sole exception of GATS/sem,⁴ scores for positive values of α are significantly greater ($p < 10^{-30}$, common language effect size: $0.6902 \leq f \leq 0.9993$). Training boosting trees only on models with $\alpha > 0$ degrades the fit (RMSE = 0.26 on the test set) but also redistributes the importance of the predictors, with tasks playing a dominant role (0.44 mean absolute SHAP) and α remaining a close second (0.37 mean absolute SHAP).

These different optimal settings strongly suggest that models that fare well on traditional NLP tasks likely do not dominate on morphology tasks. To establish whether this expectation is borne out, we compute the correlation of scores for each pair of tasks. Given the low scores for negative values of α we have established, we restrict our observations to values of $\alpha > 0$; we refer the reader to Appendix C.4 for related results across all models. Results are shown in Figure 1. We observe that NLP benchmarks (all subsets of GATS, SimLex-999, FEEL) correlate well with each other, but less well with the morphological tasks we devised (SCC, PCC, SCP, PCP for inflection; DerCF, DerCS, DerPF and DerPS for derivation), where correla-

⁴In fact, for GATS/sem, we find the opposite trend: Higher values of α lead to lower scores ($p < 10^{-5}$, $f = 0.4266$). This is due to the fact that the highest values of α lead to even greater decreases in performance than the lowest values of α . Values of $\alpha \in \{-0.2, 0.2\}$ yield the highest scores.

tion is lower and occasionally even negative. With the exception of DerCS, derivational tasks pattern in the middle, being highly correlated with each other, and having middling levels of correlation with both inflectional and semantic tasks.

5 Discussion

Two types of distributional information. The results we observe in Section 4 suggest that models exhibit a range of behaviours between two poles, defined by whether the task is testing knowledge of semantics or of morphosyntactic properties. Performances on inflectional morphology on the tasks we devised were uncorrelated or even anticorrelated to SimLex-999 and GATS/sem results, the tasks that targets lexical semantics in the most narrow sense in our set. Why is that we observe such extreme trade-offs—where better performances on semantic similarity entail lower performances on inflection, with derivation and POS-tagging patterning in the middle? One possible answer lies in the theoretical framework underpinning static word embeddings such as word2vec, i.e., distributional semantics.

As Sahlgrén (2008) and Gastaldi (2021) outline, the distributional semantics framework of Harris (1954) has historical ties with linguistic structuralism, through the works of Bloomfield (Bloomfield, 1933) and indirectly those of de Saussure (de Saussure, 1916). If we consider the objective function of neural embeddings such as word2vec, we see that these models broadly attempt to predict a target word given its context: Embeddings attempt to capture a conditional probability $p(t|c)$ of targets t given their context of occurrence c .⁵ This is the hallmark of a “paradigmatic model”, as Sahlgrén (2008) puts it: In short, these models are trained to guess which word might appear in a given context. To hearken back to linguistic structuralism, we can say these models attempt to fill in a given paradigmatic slot in a syntagmatic context, or that they try to establish associative series—which can involve either formal relations or semantic relations.

From a distributionalist point of view, contexts of occurrence constrain words in two different manners: through morphosyntactic dependencies and through lexical semantic requirements. In fact, these two different types of constraints are obvious if we compare the following examples:

- (i) You know, this is the way we eat in _____.
- (ii) I think this game is really _____.

One can easily surmise that the blanked word in Example (i) has to refer to a place: In other words, the distributional cues around this gap constrain the lexical semantics of words that can fit this specific context. On the other hand, Example (ii) leaves the semantics very much unconstrained, but requires specific morphosyntactic features—valid inserts range from “easy” to “stupid” to “dark” but their validity hinges on their adjectival nature.

If we now return to our embeddings evaluation, we can observe that the two different types of distributional constraints entail that it is logically possible that some tasks may show uncorrelated behavior, as they measure a model’s ability to capture one or the other of these types of constraints. Success on our inflectional tasks requires a proper modeling of morphosyntactic cues, whereas success on SimLex-999 requires a proper modeling of lexical semantics. These two sets of tasks are extreme positions in a trade-off situation: for SimLex-999, morphosyntactic cues are irrelevant; likewise for our inflectional tasks, capturing lexical semantics is much less important—and may actually be detrimental to performance. That these two sets of tasks correspond to extreme positions does suggest that most distributional representation evaluations tasks can be classified along two continuums, depending on the extent to which they probe lexical semantics and morphosyntactic modeling. These two aspects are not orthogonal, but it is nevertheless useful to consider them as distinct—especially given the intermediate position of derivational tasks and POS tagging, as shown by their optimal α values (Appendix C.1) and correlation patterns (Figure 1).

Derivation in the middle. Derivational tasks inherently rely on a combination of morphosyntactic and lexical semantics knowledge: French deverbal nouns in *-eur* can denote human professions (*recruteur* ‘recruiter’), properties of human agents (*fumeur*, ‘smoker’) or inanimate instruments (*compteur* ‘counter’), among others. Properly handling *-eur* forms requires that models capture on the one hand the morphosyntactic regularities surrounding agent or instrument nouns (e.g. often preceded by an article, often within short distance of a transitive verb), and on the other hand the different possible relationships of lexical semantics between a verbal base and its noun in *-eur* (agent, instrument etc).

⁵In practice, word2vec models can involve the related probabilities $p(c|t)$ (for skip-gram models) or $p(t \in c)$ (for negative sampling models). Both of these can be related to the probability of interest through renormalization or Bayes’ rule.

Further strengthening our analysis of distributional constraints as morphosyntactic or semantic, we find that POS tagging, a task that is inherently about capturing morphosyntactic relationships, but which abstracts over individual relationships in order to uncover regularities of a different nature, patterns in between inflection and derivation.

Morphological clustering tasks. An important point to stress is that the clustering tasks always return negative average silhouette scores. In other words, on average, any datapoint in the SCC, PCC, DerCF and DerCS tasks could be better assigned to some other cluster. This would suggest that morphosyntactic contrasts do not shape the vector space landscape in an intuitive, meaningful way, even when the model has optimal hyperparameters for the task. This is perhaps because paradigm cells, despite their foundational role in morphological theory, need not describe a coherent group of usages: despite both being plural nouns, *chairs* refers to more than one instance of CHAIR while *scissors* refers to a singular object. This is an extreme example of a state of affairs that plagues the concept of paradigm cell. Our tasks are also defined on imbalanced classes, which intuitively makes the tasks at hand more challenging. Furthermore, the performances we observe on prediction tasks (SCP, PCP, DerPF, DerPS) are clearly above random chance or majority label heuristics:⁶ This again confirms that morphosyntactic cues are properly encoded in suitably hyperparametrized models, suggesting that poor performance of clustering tasks is a consequence of the geometry of vector spaces being defined by permeable boundaries between paradigm cells rather than a result of models failing to capture existing patterns.

This fact also explains the behavior of the DerCS task: while we can expect the information necessary for solving the task to be present in models that capture morphosyntactic features (as evidenced by DerPS), the layout of the space makes this information hard to retrieve by clustering means. In addition, the use of semantic labels also entails that the derivational relation we selected have lexical semantic correlates, which can be exploited to perform well on the task. DerCS performance would then be reliant on the same cues as semantics tasks, explaining why it unexpectedly patterns with them.

⁶Macro-F1 for majority baselines: SCP: 0.019; PCP: 0.370; DerPF 0.006; DerPS: 0.051.

The deal with GATS. Analogy solving is another case where prior assumptions are not borne out by our experiments. GATS/I and GATS/D should in principle pattern with inflectional and derivational tasks respectively—however, all GATS tasks behave more in line with semantic tasks.

One possible source of this unexpected result is the frequency of the words employed in GATS. GATS contains only fairly frequent lexemes, which are more likely to have more senses, and irregular semantic and morphological relationships to their base (Patterson et al., 2001; Baayen and del Prado Martín, 2005; Wu et al., 2019)—all of which place GATS/I further along on the lexical semantics gradient than our tasks, which contain words from all parts of the frequency gradient. GATS morphological analogies do however occupy a median position: results on I and D analogies are not as unrelated to morphological tasks as results for the semantic-type analogies.

It is also worth noting that one can trivially obtain high results on morphological analogies through linear offset methods without having to encode morphosyntactic features. If vectors only track lexical semantic distributional constraints, then we can expect two inflected forms of a given lemma to have roughly equivalent embeddings. In such a scenario, morphology-based analogies like *danse:dansait::mange:mangeait* would entail that $\vec{danse} - \vec{dansait} \approx \vec{mange} - \vec{mangeait} \approx \vec{0}$, and therefore solving these analogies through linear offsets would devolve into a trivial solution, e.g.:

$$\begin{aligned} \vec{x} &= \vec{danse} - \vec{dansait} + \vec{mangeait} \\ &\approx \vec{0} \qquad \qquad \qquad + \vec{mangeait} \approx \vec{mange} \end{aligned}$$

In other words, it is in principle possible for models that do not encode morphological traits in any relevant way (i.e., that only consider lexical semantic distributional constraints) to succeed on this supposedly morphological benchmark. Linzen (2016) raises similar concerns and stresses that cues often lie close to one another in word2vec space, which is only one of the major points for which the analogy task has been criticized (e.g. Rogers et al., 2017; Schluter, 2018; Garg et al., 2018).

Why α ? This gradient take on distributional benchmarking tasks also explains why shaping the negative sampling distribution is found to be so impactful. If what is needed to succeed on inflectional tasks is a good representation of the morphological contrasts instantiated by the language of interest,

negative evidence for learning these contrasts can be easily found at the very top of the vocabulary’s frequency list: Contrasting the word of interest with the full paradigm of a handful of frequent lexemes in the language would get one most of the way to a working representation of morphological contrasts. Such extreme selection based on frequency is not suited for semantic tasks, which benefit from having a wider variety of negative examples and thus prefer lower values of the exponent compared to more purely morphological tasks.

To take a concrete example, consider the word *is*. This word is highly frequent, and an exceptionally poor disambiguator of aptitude to continue a particular sentence: *is* can be used to express any property intrinsic to the subject or circumstantial (*she is good* vs *he is here*), to imply existence (*she thinks therefore she is*), as an auxiliary to convey the tense, aspect and mood of another verb (*he is going out*, *she is to go there tomorrow*). Because of its wide variety of uses, *is* may take any noun as its subject or object, it may be modified by several adjectives and adverbs, and may be found in a wide variety of grammatical constructions. The sheer frequency of the verb exacerbates this feature of its usage. The distributional representation of *is* will therefore collapse all of these uses into the same representation, leading to a word embedding which is itself not necessarily helpful in pinning down the meaning and usage of *is*, but which is a good representation of which cues are not particularly informative about a word’s meaning, since they may co-occur with many outcomes.

Hence we expect word frequency to be an accurate correlate of words that are poor disambiguators: Not only do frequent words by definition occur in a large amount of linguistic contexts, they also tend to have more senses (Zipf, 1942) and to occur in more varied contexts (Dennis and Humphreys, 2001). It is therefore unsurprising that disproportionately taking frequent words as negative examples is helpful for morphological tasks: because of the variety of contexts they occur in, they are going to be particularly useful in warding off unwarranted associations that are not important for creating a representation of the target word.

Furthermore, frequent words are more likely to have irregular morphology, while infrequent words are much more likely to behave regularly (Wu et al., 2019). While both regular and irregular words may be frequent, it is very rare to find infrequent irregular words: If a word does not follow regular

patterns, this information must be explicitly encoded in the mental lexicon, which is only possible if the word is frequent enough to have a sufficiently strong mental representation.⁷ Calling morphological behavior “regular” amounts to saying that the morphological pattern applies to many words, most of which will be infrequent. Conversely, one expects irregular patterns to apply to a few frequent words (Beniamine, 2018)—i.e., frequent words have more varied behavior than infrequent words. Hence, in order for a model to learn morphology, it must focus on frequent words, which are the locus of the greatest variety of patterns in the system.

This hypothesis correctly predicts that tasks in which knowledge of morphosyntactic information about specific words is being targeted will benefit from the highest values of α : in our case, inflectional tasks, closely followed by POS tagging (which targets more abstract morphosyntactic properties that aggregate over larger groups of words), followed by derivational tasks (which target morphosyntactic and lexical semantics information simultaneously) and lastly by those targeting lexical semantics alone (SimLex-999, FEEL, GATS/Sem). It also predicts that while tasks targeting lexical semantics might benefit from lower values of α , no linguistic task will benefit from oversampling from the tail of the vocabulary with $\alpha \leq 0$.

6 Conclusions

In this paper, we showed that the performance of static embeddings on morphological tasks need not correlate with their performance on lexical semantic tasks, which constitute most major NLP benchmarks. Morphological tasks can be shown to benefit from different hyperparameters than semantic tasks; optimal settings for derivational and inflectional processes also differ.

This is all the more crucial in theoretical morphology approaches aiming to use distributional representations as meaning proxies: our findings highlight that the exact hyperparametrization can affect the outcome we observe. Choosing hyperparameters is not theoretically neutral, and different conclusions may emerge from different settings. In particular, works in theoretical morphology that rely on embeddings to compare derivation and inflection (e.g. Bonami and Paperno, 2018; Rosa and

⁷Work on language change supports this statement: Words taking irregular patterns disappear from the language, or regularize, unless they are frequent enough to have their irregularity memorized (e.g. Lieberman et al., 2007).

Žabokrtský, 2019) are at risk of reporting conclusions biased in favor of inflection or derivation, depending on the exact hyperparametrization of their embeddings.

This methodological point ties in to another contribution of this work, namely that we experimentally underscore that distributional representations are not purely lexical semantic representations, but also incorporate morphosyntactic features. This contrasts with the often held position that distributional models are to be construed as meaning representations (e.g. Schütze, 1992; Lenci, 2018; Boleda, 2020; Apidianaki, 2023). The historical structuralist roots of distributionalism highlighted by Sahlgren (2008) and Gastaldi (2021) are especially useful to understand the limits inherent to this position.

Beyond theoretical remarks, this work also offers perspectives for other applications of distributional models: Applications of (contextualized) embedding architectures to morphology may have interest in manipulating the frequency of the examples shown to the model. In particular, modeling inflection benefits from paying close attention to the head of the unigram distribution of words in a corpus: We plan to explore whether sampling from different smoothed vocabulary distributions also helps models such as BERT (Devlin et al., 2019) to capture inflectional patterns more accurately.

In all, the growing number of applications of NLP to morphology makes it imperative that we think more carefully about the data and tasks we use for evaluation. Research attempting to construct tools for morphology and morphologically rich languages might be hindered by the Anglo-centric approach prevalent in NLP. Here, we have demonstrated for French CBOWs that the common practice of setting the α hyperparameter to 0.75 following Mikolov et al. (2013) is in fact inappropriate—not only for morphology modeling but also for classical NLP benchmarks. This is all the more concerning given that French is a well-documented, resource-rich language with a vibrant NLP research community, and begs the question of how inappropriate are Anglo-centric choices for typologically more distinct languages.

Acknowledgements

We thank Olivier Bonami, Denis Paperno, and the multiple anonymous reviewers for comments on this work that substantially bettered it, as well as

James Robert Jarmusch for his creative input.



This work is part of the FoTran project, funded by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation program (agreement № 771113). We also thank the CSC-IT Center for Science Ltd., for computational resources and NVIDIA AI Technology Center (NVAITC) for the expertise in distributed training.

References

- Amine Abdaoui, Jérôme Azé, Sandra Bringay, and Pascal Poncelet. 2017. *Feel: a french expanded emotion lexicon*. *Language Resources and Evaluation*, 51(3):833–855.
- Chantal Amrhein and Rico Sennrich. 2021. *How suitable are subword segmentation strategies for translating non-concatenative morphology?* In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 689–705, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Marianna Apidianaki. 2023. *From Word Types to Tokens and Back: A Survey of Approaches to Word Meaning Representation and Interpretation*. *Computational Linguistics*, pages 1–59.
- R Harald Baayen and Fermín Moscoso del Prado Martín. 2005. Semantic density and past-tense formation in three germanic languages. *Language*, pages 666–698.
- Niyati Bafna and Zdeněk Žabokrtský. 2022. *Subword-based cross-lingual transfer of embeddings from Hindi to Marathi and Nepali*. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 61–71, Seattle, Washington. Association for Computational Linguistics.
- Siamak Barzegar, Brian Davis, Manel Zarrouk, Siegfried Handschuh, and Andre Freitas. 2018. *SemR-11: A multi-lingual gold-standard for semantic similarity and relatedness for eleven languages*. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Sacha Beniamine. 2018. *Typologie quantitative des systèmes de classes flexionnelles*. Ph.D. thesis, Université Paris Diderot.
- Leonard Bloomfield. 1933. *Language*. Henry Holt.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. *Enriching word vectors with subword information*. *Transactions of the Association for Computational Linguistics*, 5:135–146.

- Gemma Boleda. 2020. [Distributional semantics and linguistic theory](#). *Annual Review of Linguistics*, 6(1):213–234.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. [Man is to computer programmer as woman is to homemaker? debiasing word embeddings](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Olivier Bonami and Matías Guzmán Naranjo. 2023. [Distributional evidence for derivational paradigms](#), pages 219–258. De Gruyter, Berlin, Boston.
- Olivier Bonami and Denis Paperno. 2018. [Inflection vs. derivation in a distributional vector space](#). *Lingue e linguaggio, Rivista semestrale*, (2/2018):173–196.
- Francis Bond and Kyonghee Paik. 2012. A survey of wordnets and their licenses. pages 64–71.
- Kris Cao and Marek Rei. 2016. [A joint model for word embedding and word morphology](#). In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 18–26, Berlin, Germany. Association for Computational Linguistics.
- Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. 2018. [Word2vec applied to recommendation: Hyperparameters matter](#). In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys ’18, page 352–356, New York, NY, USA. Association for Computing Machinery.
- Ryan Cotterell and Hinrich Schütze. 2015. [Morphological word-embeddings](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1287–1292, Denver, Colorado. Association for Computational Linguistics.
- Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. [Morphological smoothing and extrapolation of word embeddings](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1651–1660, Berlin, Germany. Association for Computational Linguistics.
- Ferdinand de Saussure. 1916. *Cours de linguistique générale*. Payot, Paris.
- Simon Dennis and Glyn W Humphreys. 2001. A context noise model of episodic word recognition. *Psychological Review*, 108(2):452–478.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- John Rupert Firth. 1957. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-59:1–32.
- Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.
- Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. [Word embeddings quantify 100 years of gender and ethnic stereotypes](#). *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.
- Juan Luis Gastaldi. 2021. [Why can computers understand natural language?](#) *Philosophy & Technology*, 34(1):149–214.
- Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuo. 2016. [Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn’t](#). In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. [Learning word vectors for 157 languages](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Matías Guzmán Naranjo and Olivier Bonami. 2021. Overabundance and inflectional classification: Quantitative evidence from Czech. *Glossa*, 6.
- Matías Guzman Naranjo and Olivier Bonami. 2023. [Distributional assessment of derivational semantics](#). Presented at the 53rd Annual Meeting of the Societas Linguistica Europaea. Bucharest, Romania.
- Zellig S. Harris. 1954. [Distributional structure](#). *Word*, 10(2-3):146–162.
- Nabil Hathout, Franck Sajous, and Basilio Calderone. 2014. GLÀFF, a large versatile French lexicon. In *Proceedings of LREC 2014*.
- Richard Huyghe and Marine Wauquier. 2020. [What’s in an agent?](#) *Morphology*, 30:185–218.
- Arne Köhn. 2015. [What’s in an embedding? analyzing word embeddings through multilingual evaluation](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Lisbon, Portugal. Association for Computational Linguistics.
- Gabriella Lapesa, Lea Kawaletz, Ingo Plag, Marios Andreou, Max Kisselew, and Sebastian Padó. 2018. [Disambiguation of newly derived nominalizations in context: A distributional semantics approach](#). *Word Structure*, 11(3):277–312.

- Alessandro Lenci. 2018. [Distributional models of word meaning](#). *Annual Review of Linguistics*, 4(1):151–171.
- Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2022. [A comparative evaluation and analysis of three generations of distributional semantic models](#). *Lang. Resour. Eval.*, 56(4):1269–1313.
- Erez Lieberman, Jean-Baptiste Michel, Joe Jackson, Tina Tang, and Martin A Nowak. 2007. [Quantifying the evolutionary dynamics of language](#). *Nature*, 449(7163):713–716.
- Tal Linzen. 2016. [Issues in evaluating semantic spaces using word analogies](#). In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 13–18, Berlin, Germany. Association for Computational Linguistics.
- Scott M Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Marco Marelli and Marco Baroni. 2015. [Affixation in semantic space: Modeling morpheme meanings with compositional distributional semantics](#). *Psychol Rev.* PMID: 26120909.
- 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*, pages 174–184.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. [Linguistic regularities in continuous space word representations](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.
- Fiammetta Namer, Nabil Hathout, Olivier Bonami, Georgette Dal, Dany Amiot, Lucie Barque, Gilles Boyé, Stéphanie Caët, Basilio Calderone, Christine Da Silva Genest, Alexander Delaporte, Guillaume Duboisdindien, Achille Falaise, Natalia Grabar, Pauline Haas, Frédérique Henry, Mathilde Huguin, Nyoman Juniarta, Loïc Liégeois, Stéphanie Lignon, Lucie Macchi, Grigoriy Manucharian, Caroline Masson, Fabio Montermini, Nadejda Okinina, Alexandre Roulois, Franck Sajous, Daniele Sanacore, Mai Thi Tran, Juliette Thuilier, Yannick Toussaint, Delphine Tribut, and Marine Wauquier. 2023. [Demonette-v2](#). April 2, 2023.
- Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. [Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures](#). Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.
- Karalyn Patterson, Matthew A Lambon Ralph, John R Hodges, and James L McClelland. 2001. [Deficits in irregular past-tense verb morphology associated with degraded semantic knowledge](#). *Neuropsychologia*, 39(7):709–724.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. [Scikit-learn: Machine learning in Python](#). *Journal of Machine Learning Research*, 12:2825–2830.
- Radim Řehůřek and Petr Sojka. 2010. [Software Framework for Topic Modelling with Large Corpora](#). In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. <http://is.muni.cz/publication/884893/en>.
- Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017. [The \(too many\) problems of analogical reasoning with word vectors](#). In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148, Vancouver, Canada. Association for Computational Linguistics.
- Rudolf Rosa and Zdeněk Žabokrtský. 2019. [Attempting to separate inflection and derivation using vector space representations](#). In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 61–70, Prague, Czechia. Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics.
- Magnus Sahlgren. 2008. [The distributional hypothesis](#). *The Italian Journal of Linguistics*, 20:33–54.
- Natalie Schluter. 2018. [The word analogy testing caveat](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 242–246, New Orleans, Louisiana. Association for Computational Linguistics.
- Hinrich Schütze. 1992. [Word space](#). In *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann.
- Pavol Štekauer. 2014. [Štekauer, P. 2014. 'Derivational paradigms.'](#) In: Lieber, R. – Štekauer, P. (eds.) *The Oxford Handbook of Derivational Morphology*. Oxford: Oxford University Press, 354-369, pages 354–369.
- Milan Straka and Jana Straková. 2017. [Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe](#). In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Matej Ulčar, Kristiina Vaik, Jessica Lindström, Milda Dailidėnaitė, and Marko Robnik-Šikonja. 2020. [Multilingual culture-independent word analogy datasets](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4074–4080, Marseille, France. European Language Resources Association.

Rossella Varvara. 2017. *Verbs as nouns: empirical investigations on event-denoting nominalizations*. Ph.D. thesis.

Rossella Varvara, Gabriella Lapesa, and Sebastian Padó. 2021. [Grounding semantic transparency in context](#). *Morphology*, 31(4):409–446.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. [Probing pretrained language models for lexical semantics](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Marine Wauquier. 2020. *Confrontation des procédés dérivationnels et des catégories sémantiques dans les modèles distributionnels*. Ph.D. thesis. Thèse de doctorat dirigée par Hathout, Nabil Sciences du langage Toulouse 2 2020.

Shijie Wu, Ryan Cotterell, and Timothy J. O’Donnell. 2019. [Morphological irregularity correlates with frequency](#). *CoRR*, abs/1906.11483.

Britta Zeller, Sebastian Padó, and Jan Šnajder. 2014. [Towards semantic validation of a derivational lexicon](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1728–1739, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

George Kingsley Zipf. 1942. The unity of nature, least-action, and natural social science. *Sociometry*, 5(1):48–62.

A Word2vec and what it means to the linguist

The first difficulty that comes to the linguist with the adoption of NLP tooling is that of understanding and interpreting the mechanics of the software at hand. In this section, we start by providing a brief technical overview of how the CBOW model of Mikolov et al. (2013) functions in Appendix A.1, and move on to a linguistics-oriented characterization of its hyperparameters in Appendix A.2.

A.1 Algorithmic overview

At their core, distributional semantics models attempt to characterize the distribution of words. For

neural-based models, this almost always entails estimating the probability of a token t in its context c :

$$\Pr(t|c) \quad (1)$$

where c corresponds to some notion of context: For CBOW, the context is construed as a sliding window of words co-occurring in a sentence; For BERT, contexts are equated to sentences; for causal language models such as GPT, the context is understood as all preceding words.

The CBOW architecture models probabilities such as Equation (1) by means of learned vector representations for words and contexts:

$$\Pr(t_i|c = (t_1 \dots t_m)) \propto \vec{t}_i \cdot \vec{c}_i \quad (2)$$

Context representations correspond to sums of word-level features:

$$\vec{c}_i = \sum_{j=\min(1, i-w)}^{i-1} \vec{e}_j + \sum_{j=i+1}^{\max(m, i+w)} \vec{e}_j \quad (3)$$

As such, the CBOW model consists in two sets of vector representations: target vectors \vec{t}_i , which are solely used for estimating the probability of a word in context, and input embeddings \vec{e}_j which serve both as a means to model the context and as input features for downstream applications. Most studies, this one included, concerns themselves with the latter embeddings.

In the specific implementation we rely on (viz. the gensim implementation, Řehůřek and Sojka, 2010), the window size w can be either fixed or stochastically determined for every training example. In details, this sampling corresponds to replacing the window size w in Equation (3) with an effective window size \hat{w} uniformly sampled between 1 and w :

$$\hat{w} \sim U(1, w) \quad (4)$$

In practice, this entails that context words that are $k \leq w$ tokens apart from the target word are discarded in k/w of the training instances. Therefore, context words that are closer to the target word are more likely to be taken into account for prediction.

To estimate what probability to assign for a given token in a given context, a practical approach consists in training the model using both positive and negative evidence, through a procedure known as “negative sampling.” This is equivalent to maximizing the objective \mathcal{O} listed in Equation (5):

$$\mathcal{O} = \Pr(t_i|c) - \sum_{t_n \in N} \Pr(t_n|c) \quad (5)$$

Simply put, a negative-sampling CBOW model is trained to maximize the likelihood of an attested word t_i in its context c , and minimize the likelihood of all words in a set N of negative examples (not attested in this context c). The negative examples $t_n \in N$ are randomly sampled for each positive examples, using a distribution derived from the raw frequency distribution $\Pr(t)$ of word types t in the training corpus:

$$p_n(t) \propto \Pr(t)^\alpha \quad (6)$$

As is usual with neural networks, the parameters $\vec{t}_1 \dots \vec{t}_V$ and $\vec{e}_1 \dots \vec{e}_V$ are estimated through stochastic gradient descent, with the goal of maximizing the objective \mathcal{O} in Equation (5). Rather than testing for convergence of this objective on a held-out validation set, it is more usual to expose all of the available training data to the model for a pre-determined number of times, or ‘epochs.’

A.2 Interpretation of hyperparameters

A keen reader, having suffered through Appendix A.1, might notice that the algorithm of a CBOW negative-sampling model is—at least in part—linguistically interpretable. In the present paper, we specifically discuss five hyperparameters.

The *window size* w controls how contexts of occurrences are modeled. A large window entails that more word tokens intervene in the definition of a context representation \vec{c}_i , whereas a smaller window narrows the relevant context to the more immediate surrounding of the target word. Likewise, whether or not to employ a *dynamic window size sampling* algorithm, as detailed in Equation (4) also interest the linguist, as this window size sampling is equivalent to assigning a greater weight to context words closer to the target words. In other words, to re-purpose Firth’s (1957) famous quip, the window w controls what company a word keeps.

The *number of negative examples*, $\#N$, determines how to weigh positive and negative evidence. As a consequence, a larger sample set N of negative examples entails that the model will be more penalized for assigned non-negligible probability mass to negative evidence. Too large a N can however lead to a detrimental effect, as the model could be incentivized to focus solely on minimizing the negative evidence, thereby leading to an incoherent modeling of the positive evidence. In short, the size of the negative sample establish a position in

a trade-off between ensuring that spurious associations between negative examples and attested contexts do not arise (when $\#N$ is large), and emphasizing the importance of fitting to the attested data (when $\#N$ is small).

A related point that will interest the linguist concerns how to sample negative evidence; as we detailed in Equation (6), the CBOW architectures provide a *negative sampling smoothing hyperparameter* α to control this sampling process. Setting a value closer to 0 allows one to sample more from the tail of the vocabulary’s frequency distribution. More precisely, remark that $\alpha = 0$ entails sampling negative examples uniformly over the entire vocabulary sorted by frequency; $\alpha = 1$ matches the unigram frequency distribution in corpus; $\alpha > 1$ over-emphasizes frequent words, and $\alpha < 0$ over-emphasises infrequent ones.

Lastly, an import hyperparameter to consider is the *number of epochs*: Given that this controls how often the same positive evidence is used to adjust the model’s parameters, it has natural implications for the reach of any claim derived from the use of a CBOW model. From a practical point of view, we also remark that a lower number of epochs might result in a model that does not properly capture all the intricacies of the positive evidence used for its training—whereas a higher number of epochs can lead to a model that “over-fits” its training data, i.e., does not generalize properly to novel data.

Remark that we have ignored some key hyperparameters that are often discussed in the NLP literature. In particular, we do not discuss the dimension of the trained embedding as it has no obvious simple linguistic interpretation.

B Data used in experiments

B.1 Analogical relations in GATS

Subset	Section
Inflection	gram3-present-participle
Inflection	gram4-past-participle
Inflection	gram5-plural
Inflection	gram6-nationality-adjective
Inflection	gram7-past-tense
Inflection	gram8-plural-verbs
Derivation	gram1-adjective-to-adverb
Derivation	gram2-opposite
Semantic	antonyms-adjectives

(Continued on next column)

(Continued from previous column)

Subset	Section
Semantic	capital-common-countries
Semantic	capital-world
Semantic	city-in-state
Semantic	currency
Semantic	family

Table 3: Analogical relations in GATS, grouped as inflection, derivation or semantics.

B.2 Morphological processes in Demonette

Type	Process	N. pairs
Sem.	1A>N	1324
Sem.	1A>V	423
Sem.	1N>A	3870
Sem.	1N>V	2631
Sem.	1V>A	2960
Sem.	action	7520
Sem.	agent	2302
Sem.	el:N>A	279
Sem.	eur:V>A	356
Sem.	ième:NUM>A	57
Form.	CONVERSION:N>A	182
Form.	CONVERSION:N>V	2353
Form.	CONVERSION:V>N	2345
Form.	PST.PART:V>A	317
Form.	Vble:V>A	324
Form.	age:V>N	1625
Form.	aire:N>A	424
Form.	al:N>A	449
Form.	ance:V>N	95
Form.	ant:V>A	915
Form.	el:N>A	279
Form.	erie:A>N	99
Form.	erie:V>N	85
Form.	eur:V>A	356
Form.	eur:V>N	1580
Form.	euse:V>N	526
Form.	eux:N>A	402
Form.	ien:N>A	98
Form.	ier:N>A	201
Form.	if:N>A	372
Form.	if:V>A	132
Form.	ifier:A>V	50
Form.	ion:V>N	1946
Form.	ique:N>A	1742
Form.	iser:A>V	373

(Continued on next column)

task	α	e	N	w	s
SimLex-999	0.2	1	15	15	False
FEEL	0.2	5	25	5	False
GATS/sem	0.2	5	25	10	False
GATS/D	0.2	3	25	15	False
DerCF	0.6	3	10	5	False
DerCS	0.2	3	20	5	False
DerPF	0.6	5	5	5	False
DerPS	0.6	5	5	5	False
POS	1.0	5	10	5	False
GATS/I	0.2	5	25	10	False
PCC	1.4	3	25	5	True
SCC	1.4	3	10	5	False
PCP	1.0	5	5	5	False
SCP	1.0	5	20	10	False

Table 5: Hyperparameters of best performing model by task.

(Continued from previous column)

Type	Process	N. pairs
Form.	iser:N>V	278
Form.	itude:A>N	62
Form.	ité:A>N	1082
Form.	ième:N>A	57
Form.	ment:V>N	1285
Form.	rice:V>N	196
Form.	té:A>N	81
Form.	ure:V>N	73
Form.	é:V>A	1272
Form.	ée:V>N	66

Table 4: Processes from Démonette

C Supplementary results

C.1 Optimal hyperparameters for each task

We provide the optimal hyperparameters for each task in Table 5 for replication purposes. As noted in the main text, the most obvious trend we can identify is the α hyperparameter. We can also remark that most task benefit from training across multiple epochs (with the exception of SimLex-999), and most do not benefit from the shrinking s (with the exception of SCC). Also worth highlighting is that we do not observe that large windows favor semantic tasks.

task	highest	deciles								
	score	9 th	8 th	7 th	6 th	5 th	4 th	3 rd	2 nd	1 st
SimLex-999	0.310	0.292	0.282	0.272	0.261	0.250	0.242	0.232	0.223	0.213
FEEL	0.399	0.378	0.372	0.366	0.359	0.348	0.333	0.302	0.266	0.225
GATS/S	0.330	0.283	0.247	0.229	0.200	0.173	0.153	0.130	0.101	0.071
GATS/D	0.176	0.132	0.114	0.098	0.086	0.073	0.060	0.039	0.019	0.008
DerCS	-0.005	-0.012	-0.015	-0.019	-0.026	-0.034	-0.050	-0.080	-0.099	-0.112
DerCF	-0.026	-0.052	-0.062	-0.071	-0.083	-0.098	-0.135	-0.170	-0.193	-0.210
DerPF	0.549	0.502	0.486	0.471	0.456	0.425	0.355	0.272	0.206	0.157
DerPS	0.746	0.700	0.686	0.674	0.658	0.621	0.523	0.425	0.340	0.275
POS	0.744	0.716	0.704	0.694	0.680	0.651	0.587	0.524	0.462	0.392
GATS/I	0.379	0.332	0.305	0.284	0.259	0.237	0.204	0.183	0.158	0.119
SCC	-0.101	-0.168	-0.203	-0.251	-0.303	-0.331	-0.346	-0.354	-0.360	-0.373
PCC	-0.099	-0.157	-0.188	-0.221	-0.261	-0.292	-0.304	-0.313	-0.320	-0.329
SCP	0.817	0.778	0.752	0.719	0.600	0.434	0.374	0.325	0.273	0.228
PCP	0.526	0.486	0.475	0.464	0.449	0.403	0.394	0.391	0.389	0.387

Table 6: Maximum and deciles of scores per task

C.2 Highest performances per task

In Table 6, we summarize our models’ scores on each of the task, by looking at both the maximum score achieved and deciles. We can make two key observations: First, as stressed in the main text scores for morphological clustering tasks are systematically negative, meaning that embeddings do not form homogeneous, well-delineated clusters according to morphological features. Second, the spread between the first and ninth deciles tends to be much more extreme with morphological tasks (both inflectional and derivational) than with semantic task. Whether these results suggest that morphological distinctions are not adequately captured by distributional models in general, or whether the blame is to be pinned on word2vec more specifically is an intriguing question we intend to pursue in future work.

C.3 Correlation matrices by values of α

We can visualize the difference of quality induced by the α hyperparameter. can be visualized by plotting, for each pair of task, how individual model scores relate to one another and what value of α they use, as shown in Figure 2 for five of the tasks (simlex, DerPS, PCP, POS, GATS.D and GATS.I). Correlation in performance across pairs of tasks tends to be monotonic between our morphological tasks as well as between traditional NLP benchmarks, however our morphological tasks do not appear to align well with traditional benchmarks. The sole exception to that is the POS-tagging task,

which is found to correlate very strongly with our morphological derivation prediction tasks (shown in Figure 2i) and entertains a complex, non-linear relationship with all other NLP benchmarks. The α hyperparameter also accounts for much of the variation we observe: different values of α tend to produce easily delineated clusters of models, except when comparing GATS and SimLex-999 (see Figures 2j and 2k). In this latter case, note that values of α produce poorer results on both benchmarks the further away they stray from the optimal value of $\alpha = 0.2$, suggesting that here as well α determines much of the attested behavior.

C.4 Trends when including $\alpha < 0$

There are some interesting trends that emerge from looking at models with $\alpha < 0$ which we have not discussed in the main text so as to focus our argument on more successful models.

One interesting empirical approach that we can take to highlight the effect of these negative α hyperparametrizations consists in performing clustering analyses as shown in Figures 3a and 3b: inflectional, derivational and semantic tasks reliably clustered closely with tasks of the same linguistic type but, depending on the specific clustering algorithm, derivational tasks formed superclusters with inflection (e.g. full linkage clustering) or with semantics (e.g. UPGMA), confirming the intermediate status of derivational tasks. This matches with the argument we lay out in Section 5. However, if we instead only focus on $\alpha > 0$ as in Figures 3c and 3d, this effect is no longer observed, and deriva-

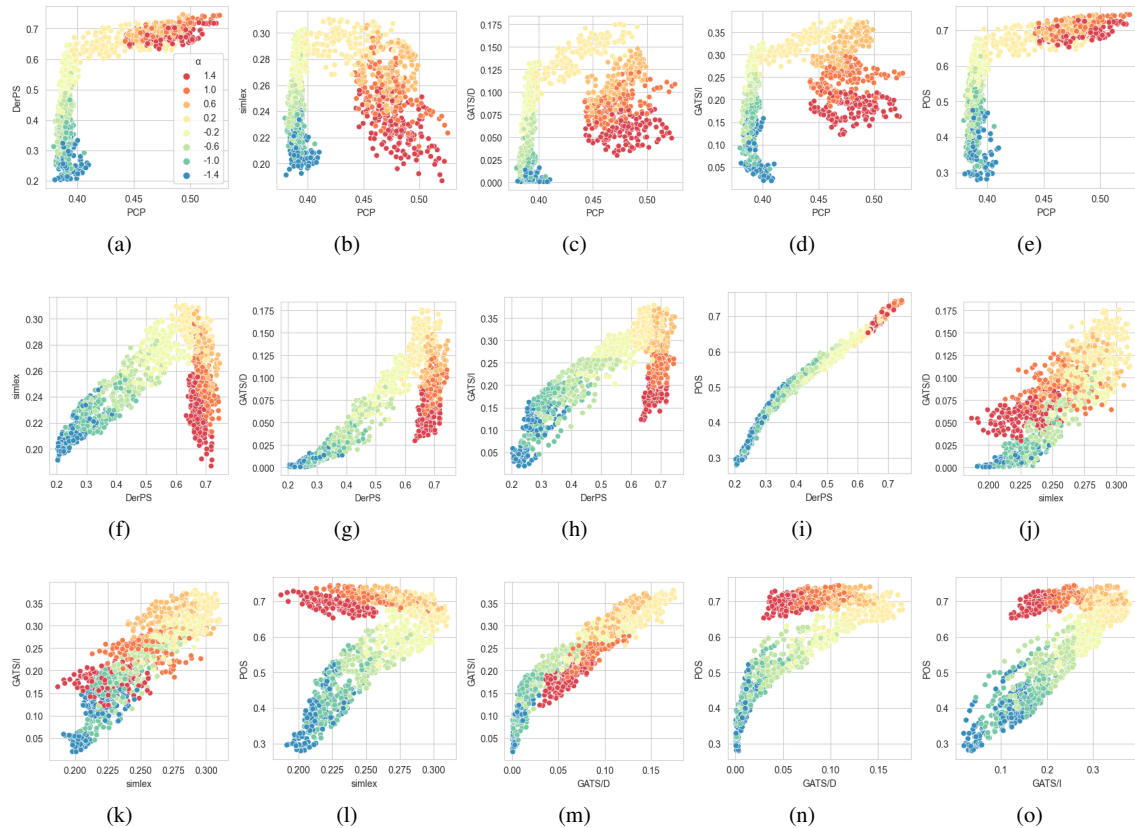
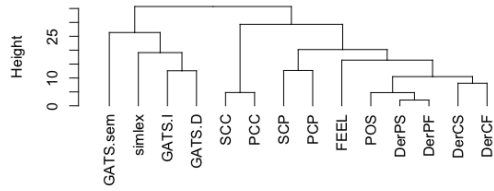


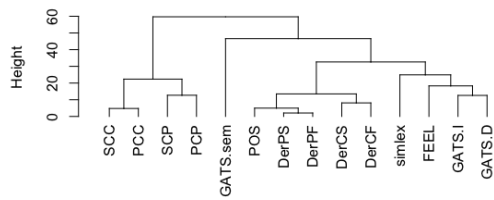
Figure 2: Selected examples of the correlation patterns found in our task set. α can be seen to account for most variation in performance.

tion tasks (with the exception of DerCS) always cluster with inflection tasks.

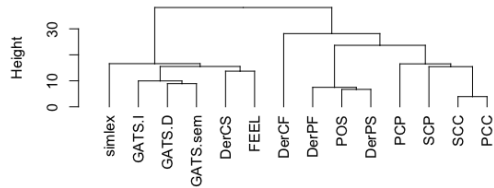
Another factor to point to is that the core observations from Section 4 also hold when looking at all models. For instance, that tasks cluster depending on the type of linguistic knowledge they target is reflected in Figure 4, although the general picture is overall less clear than when restricting the analyses to $\alpha > 0$.



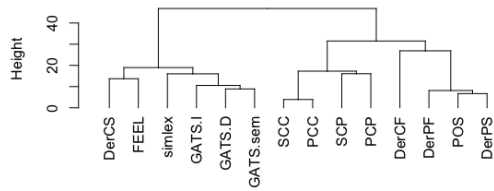
(a) UPGMA, all α



(b) Complete linkage, all α



(c) UPGMA, $\alpha > 0$



(d) Complete linkage, $\alpha > 0$

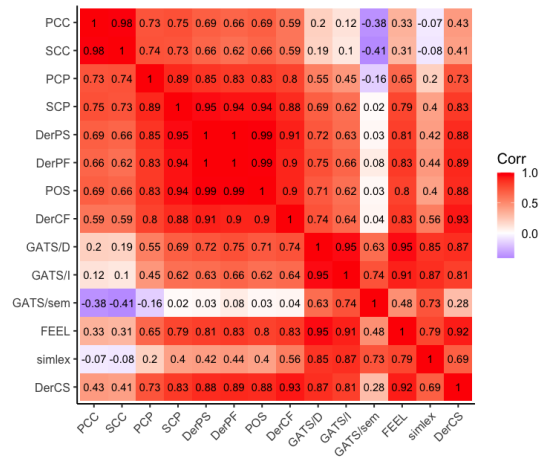


Figure 4: Spearman correlation for performance of models with all values of α on the different tasks.

Figure 3: Task hierarchical clustering based on observed scores.

How many maximum entropy grammars are predicted by a constraint set when we ignore small differences among grammars?

Giorgio Magri

CNRS, SFL, University of Paris 8 / 59 rue Pouchet, 75005 Paris, France
magrigrg@gmail.com

Abstract

All constraint-based probabilistic phonological typologies considered in the recent literature consist of uncountably many different grammars. Yet, what if two grammars that differ only slightly are coarsely counted as only one grammar when assessing the finiteness of a probabilistic typology? This paper formalizes various notions of coarse identity between probabilistic grammars and corresponding notions of coarse finiteness. It then shows that typologies of maximum entropy grammars are stubbornly infinite even when their grammars are counted coarsely (and even when the constraint set is simple, in the sense that the corresponding categorical harmonic grammar typology is finite). A companion paper shows that typologies of noisy or stochastic harmonic grammars are instead always coarsely finite (as long as the constraint set is simple). Coarse finiteness thus provides further evidence that maximum entropy is a richer, less restrictive framework.

1 Introduction

Probabilistic phonological grammars assign probabilities to phonological mappings. Probabilities take continuous values between zero and one. Hence, a probabilistic typology can contain uncountably many probabilistic grammars when we count two grammars as different in the standard sense, namely as soon as they assign different probabilities to some mapping, no matter how small the difference between those probabilities.

What if we instead tolerate some differences among probabilities as negligible? What if we count grammars **coarsely** because we count two different grammars that have only negligible differences as only one grammar? Do uncountably infinite probabilistic typologies turn finite when we count their grammars coarsely? Section 2 formalizes coarse identity between probabilistic grammars in a couple of different ways.

According to one formalization, two **ϵ -identical** probabilistic grammars can assign different probabilities to the same mapping, as long as the difference is negligible because smaller than some threshold ϵ (in absolute value). Equivalently, the ℓ_∞ distance between the two grammars is smaller than ϵ . This definition can be generalized by replacing the ℓ_∞ distance with other measures of the difference between probabilistic grammars, such as the ℓ_1 distance and the KL and χ^2 divergences.

According to another formalization of coarse identity, two **order-identical** probabilistic grammars can assign different probabilities to the same mapping as long as the difference is negligible because it does not affect the predicted probability inequalities. In other words, a mapping has a larger probability than another mapping according to one of the two grammars if and only if the same inequality holds according to the other grammar.

These notions of coarse identity yield corresponding notions of coarse finiteness. A probabilistic typology is called **ϵ -finite** or **order-finite** when it contains only finitely many grammars when we count two ϵ -identical grammars or two order-identical grammars as only one grammar. Section 3 investigates the coarse finiteness of typologies of maximum entropy (ME; Hayes and Wilson 2008) grammars.

Obviously, ME typologies always contain uncountably many grammars when we count two grammars as different in the standard sense, namely as soon as they assign different probabilities to some mapping. That is the case even when we consider only a handful of phonological mappings, no matter the constraints employed. Indeed, ME typologies are parametrized by uncountably many weight vectors and any two different weight vectors yield two ME grammars that differ because they assign different probabilities. Let us now turn from standard to coarse infinity.

To start, we consider the case of finitely many

phonological mappings. In this case, it is straightforward to verify that ME typologies are ϵ -finite and order-finite, no matter the choice of the constraints. Thus, we focus on the case of infinitely many phonological mappings, say the mappings corresponding to all underlying strings of finite but arbitrary length that can be constructed out of a finite alphabet of segments. Do ME typologies remain coarsely finite also in this case, no matter the choice of the constraints? Or can we construct counterexample constraints whose corresponding ME typologies contain infinitely many grammars even when we count grammar coarsely?

This paper shows that, for every threshold $\epsilon < 1$, it is possible to construct counterexample constraints such that the corresponding ME typology is ϵ -infinite. To illustrate, even if we choose $\epsilon = 0.999$ and are therefore willing to ignore pretty much all differences among probabilistic grammars, it is possible to construct a counterexample ME typology that is so so rich to qualify as infinite even at this level of coarseness.

Crucially, this richness is intrinsic to the ME mode of constraint interaction and does not require particularly complex constraint violation profiles. Indeed, the counterexample constraints can be chosen so simple that the corresponding categorical HG typology consists of a single grammar.

Furthermore, this result is robust: it does not depend on the specific way we measure differences among probabilities to adjudicate whether they are smaller than ϵ . Indeed, this result holds no matter whether ϵ -identity between probabilistic grammars is defined in terms of the ℓ_∞ distance or other measures of the difference between grammars, such as the ℓ_1 distances and the KL and χ^2 divergences.

Finally, this result extends from ϵ -identity to order-identity. Indeed, it is possible to construct counterexample constraints that are so simple that the corresponding categorical HG typology consists of a single grammar and yet the ME typology is order-infinite: its grammars order the infinitely many mappings made available by the phonological domain in infinitely many different ways.

The proofs of these results on ME coarse infiniteness consist of straightforward linear algebra manipulations detailed in the final appendix. The counterexample constraints constructed in these proofs are abstract and do not admit any readily available phonological interpretation. Although abstract, these counterexamples have substantial implications for the comparison between ME versus

noisy or stochastic HG (SHG; Boersma and Pater 2016; Hayes 2017; Magri and Anttila in preparation), along the following lines.

ME and SHG look *prima facie* as very similar probabilistic extensions of categorical HG. They share the formalism of weighted constraints and have been shown to make very similar empirical predictions on a variety of test cases (Hayes 2017, Flemming 2021, and Breiss and Albright 2022, among others). Alderete and Finley (2023) indeed submit that ME and SHG “make use of relatively similar mathematical foundations, and often have very similar predictions. [...] [They] produce very similar results, raising questions about what can be learned from different versions of Harmonic Grammar when the results are relatively similar. [...] It can be a challenge to compare differences between versions of Harmonic Grammar because they are so similar.”

Yet, when we look beyond empirical predictions on a simple test cases and dig deeper into the underlying mathematics, we see that SHG and ME have very different formal properties. Coarse finiteness is indeed one of the mathematical properties on which ME and SHG come apart. In fact, Magri and Anttila (in preparation) show that SHG typologies are always ϵ -finite and always order-finite, no matter the number of mappings considered, as long as the constraints are simple, in the sense that the corresponding categorical HG typology consists of only finitely many categorical grammars, which is usually the case (Pater 2009, 2016).

In other words, in the case of SHG, it is impossible to construct some counterexample constraints like those constructed here for ME, that yield an unrestricted probabilistic typology (coarsely infinite) but the most restrictive categorical HG typology (a singleton). As summarized in the concluding section 4, the results on ME coarse infinity obtained in this paper show that ME is a richer, less restrictive probabilistic extension of categorical HG than SHG is.

2 Coarse finiteness

This section develops coarse notions of finiteness for probabilistic typologies that ignore “small” differences among probabilistic grammars.

2.1 Underlying and surface forms

A **phonological mapping** is a pair (x, y) consisting of an underlying form x and a surface realization y .

The description of the phonological system of interest starts by listing into a **phonological domain** \mathcal{D} all the relevant phonological mappings. $B_{\mathcal{D}}$ denotes the **base set** of underlying forms listed by the phonological domain \mathcal{D} . And $\mathcal{D}(x)$ denotes the set of **candidate** surface realizations listed by \mathcal{D} for that underlying forms x .¹

To circumvent the problem of defining probabilities on infinite sets, a candidate set $\mathcal{D}(x)$ is usually assumed to be finite (but see Daland 2015). The base set $B_{\mathcal{D}}$ is instead allowed to be countably infinite, say because it lists all the strings of finite but arbitrary length that can be constructed out of a finite alphabet of segments.

To illustrate, the phonological domain \mathcal{D} in figure 1 consists of the sixteen phonological mappings constructed out of the four strings CV, CVC, V, and VC, that differ for whether the onset or the coda are filled or empty. The base set $B_{\mathcal{D}}$ consists of the underlying forms /CV/, /CVC/, /V/, /VC/. All candidate sets list the surface forms [CV], [CVC], [V], [VC].

2.2 Grammars and typologies

A **probabilistic (phonological) grammar** G assigns to each mapping (x, y) listed by the phonological domain \mathcal{D} a non-negative number $G(y|x) \geq 0$. We interpret this number as the probability of realizing the underlying form x as the surface candidate y . In order for this interpretation to make sense, these numbers $G(y|x)$ must be **normalized** across candidate sets, as stated in (1).

$$\sum_{y \in \mathcal{D}(x)} G(y|x) = 1 \quad (1)$$

Equivalently, a probabilistic grammar G assigns to each underlying form x in the base set $B_{\mathcal{D}}$ a **probability histogram** $G(x)$ on the corresponding candidate set $\mathcal{D}(x)$. This reformulation highlights the fact that a probabilistic grammar G only models the probability of a surface realization y of a given underlying form x , as made explicit by the notation $G(y|x)$ for **conditional probability**. A probabilistic grammar G does not model the probability of the underlying form x itself.

To illustrate, figure 2 provides two probabilistic grammars G_1 and G_2 for the phonological domain \mathcal{D} in figure 1. Grammar G_1 takes, say, the underlying form /CV/ and returns the leftmost probability histogram $G_1(/CV/)$ over the candidate set $\mathcal{D}(/CV/)$.

¹In the realm of OT, \mathcal{D} is notated *Gen*. I have changed notation to underscore the generality of the discussion.

This probability histogram assigns to the surface candidate [CV] the probability 0.6.

Finally, a **probabilistic (phonological) typology** \mathcal{T} is a collection of probabilistic phonological grammars for the same phonological domain \mathcal{D} . Throughout this section, we ignore how exactly typologies and grammars are defined (as ME grammars, as SHG grammars, and so on). The crucial point is that, no matter the choice of the framework, a probabilistic typology \mathcal{T} usually contains uncountably many different probabilistic grammars when two probabilistic grammars are counted as different in the standard sense, namely as soon as they assign slightly different probabilities to some mapping. The rest of this section thus develops coarser notions of identity between probabilistic grammars and spells out the corresponding coarser notions of finiteness for probabilistic typologies.

2.3 ϵ -finiteness

Given a threshold $\epsilon \geq 0$, two probabilistic grammars G_1 and G_2 are called **ϵ -identical** provided they assign to every mapping (x, y) in the phonological domain \mathcal{D} two probabilities $G_1(y|x)$ and $G_2(y|x)$ that differ by at most ϵ (in absolute value). To illustrate, the grammars G_1 and G_2 in figure 2 are not identical in the standard sense because, say, they assign different probabilities 0.6 and 0.55 to the mapping (/CV/, [CV]). Yet, these probabilities 0.6 and 0.55 differ by only $\epsilon = 0.05$. Analogous considerations hold for all mappings in the phonological domain \mathcal{D} . These grammars G_1 and G_2 are therefore ϵ -identical with $\epsilon = 0.05$. If we ignore differences between probabilities up to $\epsilon = 0.05$, we can count these two probabilistic grammars as the “same” grammar.

A probabilistic typology \mathcal{T} is called **ϵ -finite** provided it contains a finite subset $T \subseteq \mathcal{T}$ such that any grammar in the typology \mathcal{T} is ϵ -identical to some grammar in T . This condition is schematized in figure 3, where the red dots represent the grammars in the finite subset T , the blue dots represent all other grammars of the typology \mathcal{T} , the lines represent ϵ -identity. In conclusion, if we ignore differences between probabilities up to ϵ , the finite subset T provides as much phonological information as the original (possibly infinite) typology \mathcal{T} .

2.4 How to choose the threshold ϵ

When $\epsilon = 0$, two grammars are ϵ -identical only if they are identical in the standard sense, namely they assign exactly the same probability to every

$$\mathfrak{D} = \left\{ \begin{array}{cccc} (/CV/, [CV]) & (/CVC/, [CV]) & (/N/, [CV]) & (/VC/, [CV]) \\ (/CV/, [CVC]) & (/CVC/, [CVC]) & (/N/, [CVC]) & (/VC/, [CVC]) \\ (/CV/, [V]) & (/CVC/, [V]) & (/N/, [V]) & (/VC/, [V]) \\ (/CV/, [VC]) & (/CVC/, [VC]) & (/N/, [VC]) & (/VC/, [VC]) \end{array} \right\}$$

Figure 1: A phonological domain for basic syllable phonology

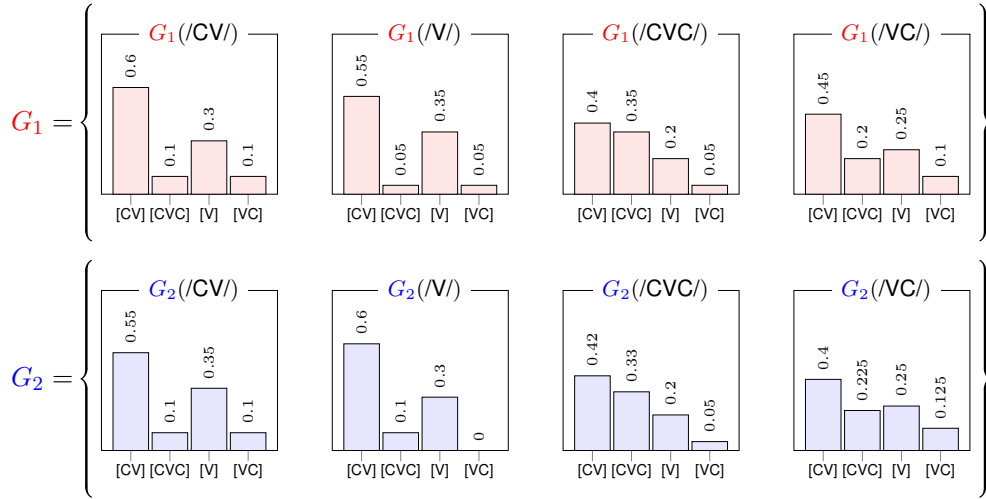


Figure 2: Two different grammars G_1 and G_2 that are nonetheless ϵ -identical with $\epsilon = 0.05$.

mapping. Hence, when $\epsilon = 0$, a probabilistic typology is ϵ -finite only if it is finite in the standard sense. In other words, ϵ -finiteness generalizes the standard notion of finiteness. As the threshold ϵ increases, we obtain coarser notions of finiteness.

When $\epsilon > 0$, the probability interval between 0 and 1 can be partitioned into finitely many disjoint intervals I_1, I_2, \dots, I_N of length at most ϵ . Suppose that the phonological domain \mathfrak{D} lists only finitely many mappings (say, because the base set $B_{\mathfrak{D}}$ lists only finitely many underlying forms and all candidate sets are finite). In this case, any probabilistic typology \mathfrak{T} is ϵ -finite because there are only finitely many ways of assigning one of the finitely many mappings from \mathfrak{D} to one of the finitely many intervals I_1, I_2, \dots, I_N . In other words, we can make infinitely many probability distinctions only when we distinguish among infinitely many mappings (namely, \mathfrak{D} is infinite) or allow arbitrarily fine grained distinctions (namely, $\epsilon = 0$).

Finally, when $\epsilon \geq 1$, any two probabilistic grammars are ϵ -identical and any probabilistic typology is therefore ϵ -finite (just choose as the subset T a singleton consisting of a unique grammar from \mathfrak{T}).

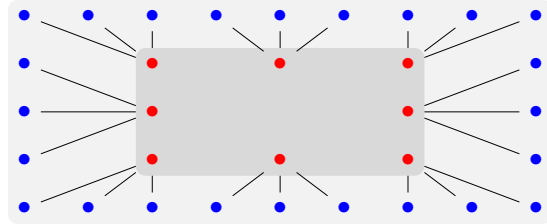


Figure 3: Schematic representation of ϵ -finiteness

In conclusion, it makes sense to investigate whether an infinite probabilistic typology \mathfrak{T} is nonetheless ϵ -finite only for ϵ between zero and one (both excluded) and when the phonological domain \mathfrak{D} lists infinitely many mappings.

2.5 Generalizing ϵ -finiteness

The notion of ϵ -finiteness introduced in subsection 2.3 can be generalized as follows. We denote by D any function that takes two probabilistic grammars G_1 and G_2 for the same phonological domain \mathfrak{D} and returns a non-negative number $D(G_1, G_2) \geq 0$ subject to the only condition that $D(G_1, G_2) = 0$ if and only if the grammars G_1 and G_2 are identical

in the standard sense, namely they assign the same probability to any mapping in the phonological domain \mathcal{D} . We will interpret the quantity $D(G_1, G_2)$ as a measure of the difference between G_1 and G_2 and thus refer to D as a **distance** between probabilistic grammars (this is a slight abuse as D need not even be symmetric: $D(G_1, G_2)$ and $D(G_2, G_1)$ can be different quantities).

Two probabilistic grammars G_1 and G_2 are then called **ϵ -identical according to D** provided their distance measured by D is at most ϵ , namely $D(G_1, G_2) \leq \epsilon$. Furthermore, a probabilistic typology \mathfrak{T} is called **ϵ -finite according to D** provided it contains some finite subset $T \subseteq \mathfrak{T}$ such that any grammar in the typology \mathfrak{T} is ϵ -identical according to D to some grammar in T . Since the distance $D(G_1, G_2)$ is equal to zero if and only if the two grammars G_1 and G_2 are identical in the standard sense, the notion of ϵ -finiteness according to D with $\epsilon = 0$ coincides with the standard notion of finiteness. In conclusion, ϵ -finiteness generalizes the standard notion of finiteness, no matter the distance D used to compare grammars.

Here is a simple strategy to define a distance between two probabilistic grammars G_1 and G_2 . First, we define a distance $D(G_1(x), G_2(x))$ between the probability histograms $G_1(x)$ and $G_2(x)$ assigned by the two grammars G_1 and G_2 to an arbitrary underlying form x in the base set $B_{\mathcal{D}}$ of the phonological domain. Then, we define the distance $D(G_1, G_2)$ between the two grammars G_1, G_2 as the largest distance between their probability histograms, as stated in (2).

$$D(G_1, G_2) = \sup_{x \in B_{\mathcal{D}}} D(G_1(x), G_2(x)) \quad (2)$$

The initial notion of ϵ -finiteness from subsection 2.3 fits into this scheme when the distance D is the ℓ_{∞} (or supremum) distance D_{∞} recalled in (3). It measures the distance between two probability histograms in terms of the largest difference between two bars for the same candidate.

$$D_{\infty}(G_1(x), G_2(x)) = \sup_{y \in \mathcal{D}(x)} |G_1(y|x) - G_2(y|x)| \quad (3)$$

Another natural distance that can be used to define ϵ -finiteness is the ℓ_1 distance D_1 recalled in (4). It measures the distance between two probability histograms in terms of the sum of the differences between two bars for the same candidates (by Scheffé's theorem, it is equal to twice the total

variation distance; see Tsybakov 2009, lemma 21, page 84).

$$D_1(G_1(x), G_2(x)) = \sum_{y \in \mathcal{D}(x)} |G_1(y|x) - G_2(y|x)| \quad (4)$$

Other natural choices for the distance D are so called f -divergences (Tsybakov 2009, section 2.4) such as the Kullback-Leibler (KL) and the χ^2 divergences. When no mapping in the phonological domain has zero probability (as is the case for ME), these two divergences are defined as in (5) and (6).

$$D_{\text{KL}}(G_1(x), G_2(x)) = \sum_{y \in \mathcal{D}(x)} G_1(y|x) \log \frac{G_1(y|x)}{G_2(y|x)} \quad (5)$$

$$D_{\chi^2}(G_1(x), G_2(x)) = \sum_{y \in \mathcal{D}(x)} \frac{(G_1(y|x) - G_2(y|x))^2}{G_2(y|x)} \quad (6)$$

2.6 From sheer sizes to inequalities

The notion of ϵ -identity looks at the sheer size of the probabilities and it is coarse because it ignores small differences in size. Various authors have suggested that we should focus not on the sheer size of the probabilities but on the inequalities they satisfy. For instance, Coetzee (2004, 2006) argues that probabilistic phonology should only model relative empirical frequencies, not absolute frequencies. In other words, a probabilistic grammar should be evaluated by comparing the inequalities among the probabilities it predicts with the inequalities among the empirical frequencies, not by fitting the predicted probabilities to the empirical frequencies.

Furthermore, the generalizations uncovered in probabilistic phonology usually consist of probability inequalities. A representative example is the famous generalization that word final t-deletion (the deletion of a stop at the end of a word preceded by another consonant) is more frequent when the following word starts with a consonant than when it starts with a vowel (see Guy 1980 and Coetzee and Kawahara 2013 for overviews). This generalization indeed consists of an inequality between the frequencies of deletion for *cost#me* versus *cost#us*. The generalization says nothing about the absolute frequencies of deletion. Indeed, Anttila and Magri (2018) and Magri and Anttila (in preparation) capture such generalizations by extending the

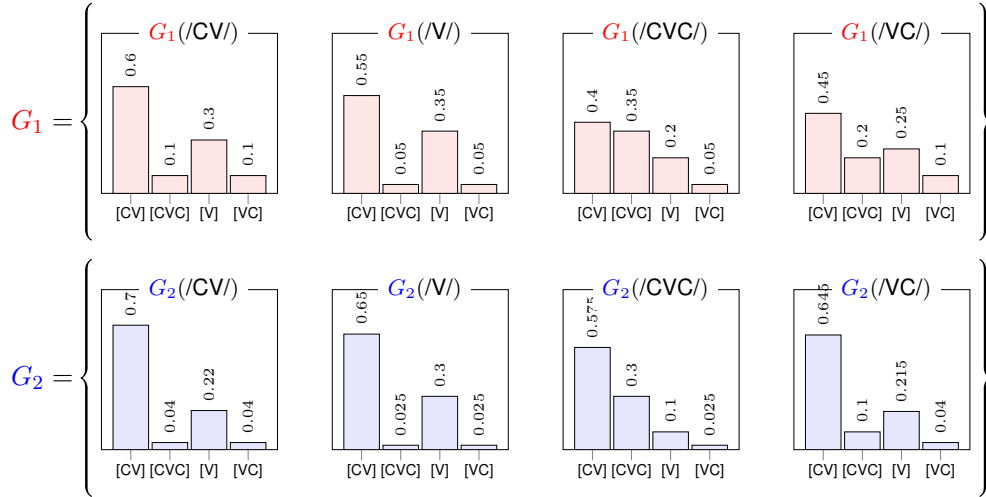


Figure 4: Two different grammars G_1 and G_2 that are nonetheless order-identical

Greenbergian implicational universals from the categorical to the probabilistic setting in terms of probability inequalities that hold uniformly across all the grammars in a probabilistic typology.

2.7 Order-finiteness

Based on these considerations, we say that two probabilistic grammars G_1 and G_2 for some phonological domain \mathfrak{D} are **order-identical** provided they agree on how they order the mappings in \mathfrak{D} in terms of the size of their probabilities: for any two mappings (x, y) and (\hat{x}, \hat{y}) from \mathfrak{D} , grammar G_1 satisfies the inequality $G_1(y|x) > G_1(\hat{y}|\hat{x})$ if and only if the other grammar G_2 satisfies the same inequality $G_2(y|x) > G_2(\hat{y}|\hat{x})$.

To illustrate, the two grammars G_1 and G_2 in figure 2 are ϵ -identical because they assign probabilities that differ by at most $\epsilon = 0.05$. Yet, they are not order-identical because these small differences in probabilities impact the inequalities. For instance, G_1 assigns more probability to $(/CV/, [CV])$ than to $(/V/, [CV])$ while G_2 does the reverse.

The situation is different for the two grammars G_1 and G_2 in figure 4. They are not ϵ -identical with $\epsilon = 0.05$ (for instance because the probabilities they assign to $(/CV/, [CV])$ differ by 0.1). Yet, both G_1 and G_2 assign more probability to $(/CV/, [CV])$ than to $(/V/, [CV])$. Analogous considerations hold for any pair of mappings in the phonological domain \mathfrak{D} : G_1 and G_2 induce the same order of the sixteen mappings according to the size of their probabilities (with ties broken in some arbitrary but fixed way), as made explicit in figure 5. We

conclude that these grammars G_1 and G_2 are order-identical. If we ignore sheer differences between probabilities and only care about the inequalities they satisfy, as argued in subsection 2.6, we can count these two probabilistic grammars G_1 and G_2 as the “same” grammar.

A probabilistic typology \mathfrak{T} is called **order-finite** provided it contains some finite set $T \subseteq \mathfrak{T}$ such that any grammar in the typology \mathfrak{T} is order-identical to some grammar in T . In other words, this finite subset T provides as much phonological information as the original (possibly infinite) typology \mathfrak{T} when we ignore sheer probabilities and only care about the inequalities they satisfy.

When the phonological domain \mathfrak{D} lists only finitely many mappings, any probabilistic typology \mathfrak{T} is order-finite, because there are only finitely many ways of ordering finitely many mappings. Thus, it makes sense to investigate whether an infinite probabilistic typology \mathfrak{T} is nonetheless order-finite only when the phonological domain \mathfrak{D} lists infinitely many mappings.

2.8 Summary

An infinite probabilistic typology is called **coarsely finite** if it is ϵ -finite relative to some distance D for some threshold ϵ between zero and one as in subsection 2.5 or order-finite as in subsection 2.7. In other words, the typology contains only finitely many grammars when we count grammars coarsely by ignoring differences between probabilities that are negligible because smaller than ϵ or because too small to affect the inequalities among probabilities.

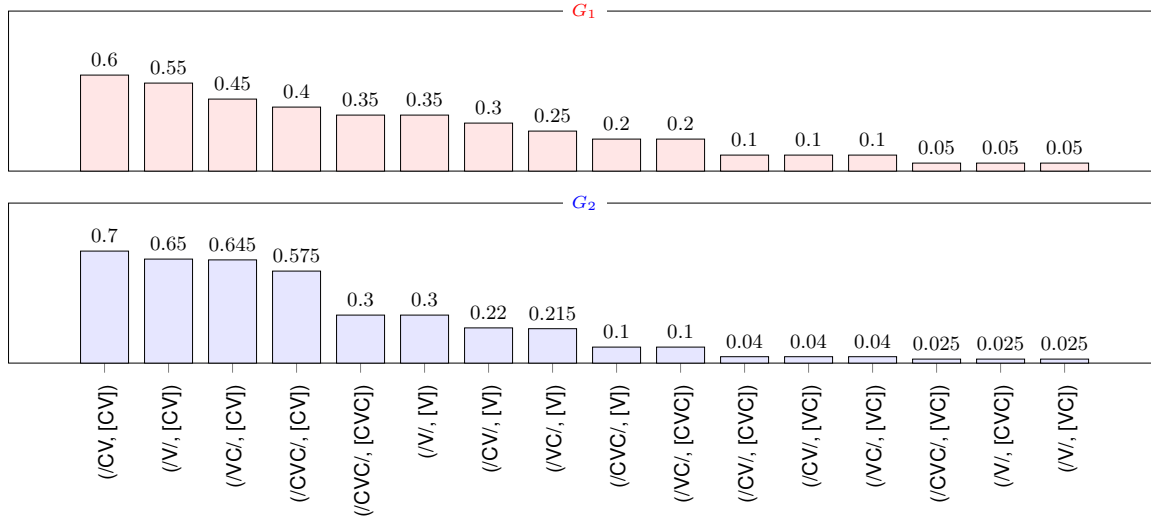


Figure 5: Grammars G_1 and G_2 in figure 4 order the mappings based on their probabilities in the same way

3 Coarse finiteness of ME typologies

This section applies these notions of coarse finiteness to the analysis of ME typologies.

3.1 Probabilistic ME typologies

So far, we have worked with arbitrary probabilistic grammars, extensionally defined as in subsection 2.2 as collections of probability histograms. Now, we focus on ME grammars, briefly recalled here. We start with a set \mathbf{C} consisting of a finite number n of phonological **constraints** C_1, \dots, C_n for the phonological domain \mathcal{D} .² A constraint C_k assigns to each phonological mapping (x, y) a number $C_k(x, y)$. This number is integral and non-negative because it is the result of counting the number of occurrences of some specific marked structure in the surface form y or the number of occurrences of some specific discrepancy between the underlying and surface forms x and y . Each constraint C_k is assigned a non-negative **weight** $w_k \geq 0$ that quantifies its importance. These weights are collected into a vector $\mathbf{w} = (w_1, \dots, w_n)$.

The probabilistic **ME grammar** $G_{\mathbf{w}}^{\text{ME}}$ corresponding to this weight vector \mathbf{w} assigns to each mapping (x, y) a probability proportional to the exponential of the opposite of the weighted sum of constraint violations, as stated in (7). The proportionality constant is univocally determined by the

normalization condition (1).

$$G_{\mathbf{w}}^{\text{ME}}(y | x) \propto \exp \left\{ - \sum_{k=1}^n w_k C_k(x, y) \right\} \quad (7)$$

The probabilistic **ME typology** defined by a phonological domain \mathcal{D} and a constraint set \mathbf{C} is the family $\mathfrak{T}^{\text{ME}}(\mathcal{D}, \mathbf{C})$ consisting of the probabilistic ME grammars (7) corresponding to all vectors \mathbf{w} of non-negative constraint weights.

3.2 Categorical HG typologies

Probabilistic ME grammars are closely related to categorical HG grammars recalled here as well. A weight vector \mathbf{w} is called **proper** provided it satisfies the following condition: for every underlying form x in the base set $B_{\mathcal{D}}$, there exists a unique surface form y (called the **winner**) in the candidate set $\mathcal{D}(x)$ that is assigned by the corresponding ME grammar $G_{\mathbf{w}}^{\text{ME}}$ a probability strictly larger than the probability assigned to each other surface form z (dismissed as a **loser**) in the candidate set $\mathcal{D}(x)$, namely $G_{\mathbf{w}}^{\text{ME}}(y | x) > G_{\mathbf{w}}^{\text{ME}}(z | x)$. The categorical **HG grammar** corresponding to a proper weight vector \mathbf{w} realizes each underlying form in the base set $B_{\mathcal{D}}$ as the corresponding winner candidate with largest ME probability. The categorical **HG typology** defined by a phonological domain \mathcal{D} and a constraint set \mathbf{C} is the family $\mathfrak{T}^{\text{HG}}(\mathcal{D}, \mathbf{C})$ consisting of the categorical HG grammars corresponding to all proper vectors \mathbf{w} of non-negative weights.

²In the realm of OT, \mathbf{C} is notated *Con*. I have changed notation to underscore the generality of the discussion.

3.3 ME typologies are not coarsely finite

Both categorical HG typologies and probabilistic ME typologies are parametrized by uncountably many weight vectors. Since categorical grammars make only binary choices, many different weight vectors yield the same categorical HG grammar. Since probabilities can instead take any continuous value between zero and one, any two different weight vectors yield two probabilistic ME grammars that are different in the standard sense, namely assign different probabilities to the same phonological mapping. As a result probabilistic ME typologies are always uncountably infinite, even when the phonological domain \mathcal{D} lists only one underlying form with only two candidate surface realizations.

Yet, in subsection 2.3 we have said that two probabilistic grammars are ϵ -identical or order-identical when the differences between the probabilities they assign are negligible because they are smaller than some threshold ϵ or they do not affect the inequalities among probabilities. We have then observed that an infinite probabilistic typology can nonetheless qualify as ϵ -finite or order-finite when indeed we count multiple ϵ -identical or multiple order-identical grammars as one single grammar.

Are ME typologies always ϵ -finite or order-finite, no matter the choice of the constraints? The following two main results provide a negative answer to this question. The proofs of these two facts consist of straightforward linear algebra manipulations detailed in the final appendix.

Result 1 *For every positive threshold $0 < \epsilon < 1$ strictly smaller than one, it is possible to construct an infinite phonological domain \mathcal{D} and a constraint set \mathcal{C} such that the corresponding ME typology $\mathfrak{T}^{ME}(\mathcal{D}, \mathcal{C})$ is ϵ -infinite while the corresponding HG typology $\mathfrak{T}^{HG}(\mathcal{D}, \mathcal{C})$ is a singleton.* \square

Result 2 *It is possible to construct an infinite phonological domain \mathcal{D} and a constraint set \mathcal{C} such that the corresponding ME typology $\mathfrak{T}^{ME}(\mathcal{D}, \mathcal{C})$ is order-infinite while the corresponding HG typology $\mathfrak{T}^{HG}(\mathcal{D}, \mathcal{C})$ is a singleton.* \square

A few remarks are in order. **(A)** Let us consider a threshold ϵ very close to one, say $\epsilon = 0.999$. This means that we are willing to ignore as negligible pretty much all disagreements among probabilities. In other words, we are willing to count as one single grammar even multiple grammars that are very different in the standard sense. And yet, even at this highest degree of coarseness, result

1 says that we can construct ME typologies that are ϵ -infinite. **(B)** This typological richness is a direct consequence of the ME mode of constraint interaction and does not require a particularly complex pattern of constraint violations. Indeed, both results guarantee that the constraints used in the ME counterexamples are very simple, in the sense that the corresponding categorical HG typology is simplest, namely consists of a single grammar. **(C)** Finally, result 1 is robust: appendix 5.3 shows that it straightforwardly extends from the original basic notion of ϵ -finiteness from subsection 2.3 to its generalization in subsection 2.5 in terms of other measures of the difference between probabilistic grammars such as the ℓ_1 distance and the KL and χ^2 divergences.

3.4 Comparison with SHG

To appreciate the significance of these results for phonological theory, we briefly turn to SHG phonology, recalled here. The probabilistic **SHG grammar** corresponding to a non-negative weight vector \mathbf{w} assigns to each mapping (x, y) a probability equal to the probability of sampling according to the normal distribution with mean \mathbf{w} some non-negative proper weight vector such that the corresponding categorical HG grammar indeed realizes the underlying form x as the surface candidate y .³ The probabilistic **SHG typology** is the family of the probabilistic SHG grammars $G_{\mathbf{w}}^{SHG}$ corresponding to all vectors \mathbf{w} of non-negative constraint weights

ME and SHG look *prima facie* as similar probabilistic extensions of categorical HG. Indeed, both ME and SHG are defined in terms of weighted sums of constraint violations. Furthermore, ME and SHG have been shown to fit equally well various patterns of empirical frequencies (Hayes 2017, Flemming 2021, and Breiss and Albright 2022, among others). Yet, SHG behaves very differently from ME in terms of coarse finiteness, as follows.

Typologies of categorical HG grammars can be infinite (Legendre et al. 2006), contrary to typologies of categorical OT grammars, that are instead always finite. Yet, OT and HG make such diver-

³Thus defined, SHG grammars can unfortunately flout the normalization condition (1): the normal distribution with mean \mathbf{w} can assign some probability to vectors that are negative or non-proper and therefore correspond to no categorical HG grammar. Hayes and Kaplan (2023) and Magri and Anttila (in preparation) discuss various modifications of the basic definition of SHG to deal with this problem. These modifications have no implications for the coarse finiteness of SHG typologies.

gent typological predictions only for very special (and possibly unwarranted) constraint configurations (Pater 2009, 2016). In general, categorical HG typologies are finite, just as OT typologies.

Magri and Anttila (in preparation) then show that, whenever the categorical HG typology corresponding to some constraint set is finite, the probabilistic SHG typology corresponding to that constraint set, although uncountably infinite, is nonetheless ϵ -finite and order-finite. It is therefore impossible to construct for SHG some counterexample constraints like those constructed here for ME, that yield a very complex probabilistic typology (coarsely infinite) but a very simple categorical HG typology (a singleton).

To illustrate, let us consider a threshold ϵ very close to zero, say $\epsilon = 0.0001$. This means that we are willing to ignore as negligible only the smallest differences among probabilities. In other words, we are willing to count as one single grammar only two grammars that are indeed very close to being identical in the standard sense. And yet, even at this lowest degree of coarseness, we cannot construct SHG typologies that are ϵ -infinite, unless we resort to special (and possibly unwarranted) constraint sets that yield infinite categorical HG typologies. We conclude that the results on ME coarse infinity obtained in this paper show that ME is a richer, less restrictive probabilistic extension of categorical HG than SHG is.

4 Conclusions

This paper has developed techniques to discretize an uncountably infinite probabilistic typology down to a finite core by ignoring small differences among probabilities. The notion of ϵ -finiteness arises when we ignore differences smaller than ϵ between the probabilities assigned by two grammars. The notion of order-finiteness arises when we ignore differences that do not compromise the inequalities among the probabilities assigned by two grammars. Magri and Anttila (in preparation) show that SHG typologies are always ϵ -finite and order-finite, as long as the constraints are simple, in the sense that the corresponding categorical HG typology is finite. This paper has shown that ME typologies can instead be ϵ -infinite and order-infinite, even when the constraints are so simple that the corresponding categorical HG typology is a singleton. We conclude that ME is a richer, less restrictive probabilistic extension of categorical HG.

References

- John Alderete and Sara Finley. 2023. Probabilistic phonology: a review of theoretical perspectives, applications, and problems. *Language and Linguistics*, 24:565–610.
- 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 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.
- Andries W. Coetzee. 2004. *What it Means to be a Loser: Non-Optimal Candidates in Optimality Theory*. Ph.D. thesis, University of Massachusetts, Amherst.
- Andries W. Coetzee. 2006. Variation as assessing ‘non-optimal’ candidates. *Phonology*, 23:337–385.
- Andries W. Coetzee and Shigeto Kawahara. 2013. Frequency biases in phonological variation. *Natural Language and Linguistic Theory*, 31(1):47–89.
- Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing.
- Robert Daland. 2015. Long words in maximum entropy phonotactic grammars. *Phonology*, 32.3:353–383.
- Edward Flemming. 2021. Comparing maxent and noisy harmonic grammar. *Glossa: a journal of general linguistics*, 6:1–42.
- Gregory R. Guy. 1980. Variation in the group and the individual: The case of final stop deletion. In William Labov, editor, *Locating language in time and space*, pages 1–36. Academic Press, New York.
- Bruce Hayes. 2017. Varieties of Noisy Harmonic Grammar. In *Proceedings of the 2016 Annual Meeting in Phonology*, Washington, DC. Linguistic Society of America.
- 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.
- G eraldine Legendre, Antonella Sorace, and Paul Smolensky. 2006. The optimality theory/harmonic grammar connection. In Paul Smolensky and G eraldine Legendre, editors, *The Harmonic Mind*, pages 903–966. MIT Press, Cambridge, MA.

Giorgio Magri and Arto Anttila. in preparation. Principles of probabilistic phonology.

Joe Pater. 2009. Weighted constraints in generative linguistics. *Cognitive Science*, 33:999–1035.

Joe Pater. 2016. Universal grammar with weighted constraints. In Joe Pater and John J. McCarthy, editors, *Harmonic Grammar and Harmonic Serialism*, pages 1–46. Equinox, London.

Alexandre B. Tsybakov. 2009. *Introduction to Nonparametric Estimation*. Springer Verlag, New York.

5 Appendices

Throughout this appendix, $\mathbf{a} \cdot \mathbf{b}$ denotes the scalar product $\mathbf{a} \cdot \mathbf{b} = \sum_{k=1}^n a_k b_k$ between two vectors $\mathbf{a} = (a_1, \dots, a_n)$ and $\mathbf{b} = (b_1, \dots, b_n)$; furthermore, $\|\mathbf{a}\|$ denotes the 2-norm $\|\mathbf{a}\| = \sqrt{\sum_{k=1}^n a_k^2}$. The proofs in this appendix consist of straightforward linear algebra manipulations.

5.1 A lemma for the proof of result 1

Lemma 1 Consider $k - 1$ vectors $\mathbf{c}_1, \dots, \mathbf{c}_{k-1}$ with positive integral components and a vector \mathbf{w}_k with positive rational components such that $\mathbf{w}_k \cdot \mathbf{c}_1 \leq 1, \dots, \mathbf{w}_k \cdot \mathbf{c}_{k-1} \leq 1$. For any $\Delta > 0$, there exist a vector \mathbf{c}_k with positive integral components and a vector \mathbf{w}_{k+1} with positive rational components such that $\mathbf{w}_{k+1} \cdot \mathbf{c}_1 \leq 1, \dots, \mathbf{w}_{k+1} \cdot \mathbf{c}_{k-1} \leq 1$ and furthermore $\mathbf{w}_{k+1} \cdot \mathbf{c}_k \leq 1$ while $\mathbf{w}_k \cdot \mathbf{c}_k \geq \Delta$.

Indeed, since the vector \mathbf{w}_k has positive rational components, it has the shape $\mathbf{w} = (\frac{a_1}{b_1}, \frac{a_2}{b_2}, \dots, \frac{a_n}{b_n})$, where $a_1, \dots, a_n, b_1, \dots, b_n$ are positive integers. Let M be the smallest common multiple of the denominators b_1, \dots, b_n . Hence, $M\mathbf{w}$ is a vector with positive integral components. We choose a positive integer $\ell > 0$ and a positive rational number $\xi > 0$ as in (8).

$$\ell \geq \frac{\Delta}{M\|\mathbf{w}_k\|^2}, \quad \xi \leq \min \left\{ 1, \frac{1}{M\ell\|\mathbf{w}_k\|^2} \right\} \quad (8)$$

We define the vector \mathbf{c}_k with positive integral components and the vector \mathbf{w}_{k+1} with positive rational components as in (9).

$$\mathbf{c}_k = \ell M \mathbf{w}_k \quad \mathbf{w}_{k+1} = \xi \mathbf{w}_k \quad (9)$$

These positions satisfy the inequalities (10) and (11) as well as the inequality (12) for every $h = 1, \dots, k - 1$, completing the proof of the lemma.

$$\mathbf{w}_k \cdot \mathbf{c}_k = \ell M \|\mathbf{w}_k\|^2 \geq \Delta \quad (10)$$

$$\mathbf{w}_{k+1} \cdot \mathbf{c}_k = \ell \xi M \|\mathbf{w}_k\|^2 \leq 1 \quad (11)$$

$$\mathbf{w}_{k+1} \cdot \mathbf{c}_h = \xi \mathbf{w}_k \cdot \mathbf{c}_h \leq \xi 1 \leq 1 \quad (12)$$

5.2 Proof of result 1

Given a threshold $0 < \epsilon < 1$, we choose two constants $0 < \epsilon_1 < \epsilon_2 < 1$ more than ϵ apart, namely $\epsilon_1 + \epsilon < \epsilon_2$. Furthermore, we choose a positive integer $m > 0$ and a positive constant $\Delta > 0$ that satisfy the inequalities in (13).

$$m \geq \frac{1 - \epsilon_1}{\epsilon_1} e, \quad \Delta \geq \log \left(m \frac{\epsilon_2}{1 - \epsilon_2} \right) \quad (13)$$

We start with an arbitrary vector \mathbf{w}_1 with positive rational components. By applying lemma 1 with $k = 1$ to this vector \mathbf{w}_1 , we conclude that there exist a vector \mathbf{c}_1 with positive integral components and a vector \mathbf{w}_2 with positive rational components that validate the red inequalities in the first step of the reasoning in figure 6. By applying again lemma 1 with $k = 2$ to the vectors \mathbf{c}_1 and \mathbf{w}_2 in the bottom line of this first step, we conclude that there exist a vector \mathbf{c}_2 with positive integral components and a vector \mathbf{w}_3 with positive rational components that validate the red inequalities in the second step of the reasoning in figure 6. By applying once again lemma 1 with $k = 3$ to the vectors $\mathbf{c}_1, \mathbf{c}_2$ and \mathbf{w}_3 in the bottom line of this second step, we conclude that there exist a vector \mathbf{c}_3 with positive integral components and a vector \mathbf{w}_4 with positive rational components that validate the red inequalities in the third step of the reasoning in figure 6. And so on and so forth.

In conclusion, we have established the existence of a sequence of vectors $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k, \dots$ with positive rational components and a sequence of vectors $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k, \dots$ with positive integral components that satisfy the k inequalities in (14) for every $k = 1, 2, \dots$

$$\begin{aligned} \mathbf{w}_k \cdot \mathbf{c}_1 &\leq 1 & \mathbf{w}_k \cdot \mathbf{c}_k &\geq \Delta & (14) \\ &\vdots & & & \\ \mathbf{w}_k \cdot \mathbf{c}_{k-1} &\leq 1 \end{aligned}$$

To construct the desired counterexample, we consider the infinite phonological domain \mathfrak{D} described in (15). For every index $k = 1, 2, \dots$, the base set $B_{\mathfrak{D}}$ of the phonological domain contains the underlying form x_k . Its candidate set $\mathfrak{D}(x_k)$ consists of $m + 1$ candidates y, z_1, \dots, z_m . For concreteness, we refer to y as the **winner** candidate and to z_1, \dots, z_m as the **loser** candidates.

$$\begin{aligned} B_{\mathfrak{D}} &= \{x_1, x_2, \dots, x_k, \dots\} \\ \mathfrak{D}(x_k) &= \{y, z_1, \dots, z_m\} \end{aligned} \quad (15)$$

Furthermore, we define the constraint set \mathbf{C} in such a way that, for every underlying form x_k and for each loser candidate z_i with $i = 1, \dots, m$, the difference between the constraint violation vector $\mathbf{C}(x_k, z_i)$ of this loser candidate minus the constraint violation vector $\mathbf{C}(x_k, y)$ of the winner candidate y is equal to the vector \mathbf{c}_k constructed in (14), as stated in (16).

$$\mathbf{C}(x_k, z_i) - \mathbf{C}(x_k, y) = \mathbf{c}_k \quad (16)$$

$$\begin{array}{ccc}
\mathbf{w}_1 \implies & \begin{array}{l} \mathbf{w}_1 \cdot \mathbf{c}_1 \geq \Delta \\ \mathbf{w}_2 \cdot \mathbf{c}_1 \leq 1 \end{array} & \implies \begin{array}{l} \mathbf{w}_1 \cdot \mathbf{c}_1 \geq \Delta \\ \mathbf{w}_2 \cdot \mathbf{c}_1 \leq 1 \\ \mathbf{w}_3 \cdot \mathbf{c}_1 \leq 1 \end{array} \\
\text{first step} & & \text{second step} \\
& & \implies \begin{array}{l} \mathbf{w}_1 \cdot \mathbf{c}_1 \geq \Delta \\ \mathbf{w}_2 \cdot \mathbf{c}_1 \leq 1 \\ \mathbf{w}_3 \cdot \mathbf{c}_1 \leq 1 \\ \mathbf{w}_4 \cdot \mathbf{c}_1 \leq 1 \end{array} \\
& & \text{third step}
\end{array}$$

Figure 6

This position (16) makes sense because the vector \mathbf{c}_k has integral components that can therefore be interpreted as differences between numbers of constraint violations under the usual assumption that constraints assign integral numbers of violations. Furthermore, the integral components of the vector \mathbf{c}_k are all positive. The position (16) thus says that every constraint in the constraint set \mathbf{C} assigns less violations to the winner mapping (x_k, y) than to each of the loser mappings (x_k, z_i) . Equivalently, the winner mapping (x_k, y) always beats each loser mapping (x_k, z_i) in HG, no matter the choice of the non-negative constraint weights. We conclude that the HG typology $\mathfrak{T}^{\text{HG}}(\mathfrak{D}, \mathbf{C})$ corresponding to the phonological domain \mathfrak{D} in (15) and the constraint set \mathbf{C} in (16) consists of a unique HG grammar, namely the grammar that realizes each underlying form x_k as its winner candidate y .

We now switch from categorical HG to probabilistic ME. We focus on the ME grammar $G_{\mathbf{w}_k}^{\text{ME}}$ corresponding to the weight vector \mathbf{w}_k in (14). We want to bound the probability it assigns to the mappings (x_h, y) with $h = 1, \dots, k-1$ and to the mapping (x_k, y) . As explained below, the inequalities $\mathbf{w}_k \cdot \mathbf{c}_h \leq 1$ with $h = 1, \dots, k-1$ on the lefthand side of (14) ensure that the ME grammar $G_{\mathbf{w}_k}^{\text{ME}}$ assigns to the mapping (x_h, y) with $h = 1, \dots, k-1$ a probability that is small, namely at most ϵ_1 , as stated in (17). Analogously, the inequality $\mathbf{w}_k \cdot \mathbf{c}_k \geq \Delta$ on the righthand side of (14) ensures that the ME grammar $G_{\mathbf{w}_k}^{\text{ME}}$ assigns to the mapping (x_k, y) a probability that is instead large, namely at least ϵ_2 , as stated in (18).

$$G_{\mathbf{w}_k}^{\text{ME}}(y | x_h) \leq \epsilon_1 \quad (17)$$

$$G_{\mathbf{w}_k}^{\text{ME}}(y | x_k) \geq \epsilon_2 \quad (18)$$

Indeed, the computation in (19) establishes the inequality (17) and an analogous computation establishes the inequality (18). Step (19a) holds because of the definition (7) of the ME probability $G_{\mathbf{w}_k}^{\text{ME}}(y | x_h)$ as proportional to the exponential of the opposite of the weighted sum of the constraint violations of the winner candidate y . The

proportionality constant is univocally determined by the normalization condition (1) and has been made explicit in the denominator. Step (19b) holds by dividing both the numerator and the denominator by $\exp\{-\mathbf{w}_k \cdot \mathbf{C}(x_h, y)\}$. Step (19c) holds because of the assumption (16) that the difference $\mathbf{C}(x_h, z_i) - \mathbf{C}(x_h, y)$ is equal to the vector \mathbf{c}_h for every $i = 1, \dots, m$. Step (19d) holds because of the assumption (14) that the scalar product $\mathbf{w}_k \cdot \mathbf{c}_h$ between the two vectors \mathbf{w}_k and \mathbf{c}_h is at most one for every $h = 1, \dots, k-1$. Finally, step (19e) holds because of the assumption that the integer m is large enough, as in (13).

$$\begin{aligned}
G_{\mathbf{w}_k}^{\text{ME}}(y | x_h) &= \quad (19) \\
&\stackrel{(a)}{=} \frac{e^{-\mathbf{w}_k \cdot \mathbf{C}(x_h, y)}}{e^{-\mathbf{w}_k \cdot \mathbf{C}(x_h, y)} + \sum_{i=1}^m e^{-\mathbf{w}_k \cdot \mathbf{C}(x_h, z_i)}} \\
&\stackrel{(b)}{=} \frac{1}{1 + \sum_{i=1}^m e^{-\mathbf{w}_k \cdot (\mathbf{C}(x_h, z_i) - \mathbf{C}(x_h, y))}} \\
&\stackrel{(c)}{=} \frac{1}{1 + m e^{-\mathbf{w}_k \cdot \mathbf{c}_h}} \\
&\stackrel{(d)}{\leq} \frac{1}{1 + m e^{-1}} \stackrel{(e)}{\leq} \epsilon_1
\end{aligned}$$

To complete the proof of result 1, we now consider the ME grammars $G_{\mathbf{w}_{k_1}}^{\text{ME}}$ and $G_{\mathbf{w}_{k_2}}^{\text{ME}}$ corresponding to two weight vectors \mathbf{w}_{k_1} and \mathbf{w}_{k_2} with $k_1 > k_2$. By (17), the ME grammar $G_{\mathbf{w}_{k_1}}^{\text{ME}}$ assigns a probability smaller than ϵ_1 to the mapping (x_{k_2}, y) , namely $G_{\mathbf{w}_{k_1}}^{\text{ME}}(y | x_{k_2}) \leq \epsilon_1$. Furthermore, by (18), the ME grammar $G_{\mathbf{w}_{k_2}}^{\text{ME}}$ instead assigns a probability larger than ϵ_2 to this mapping (x_{k_2}, y) , namely $G_{\mathbf{w}_{k_2}}^{\text{ME}}(y | x_{k_2}) \geq \epsilon_2$. Since ϵ_1 and ϵ_2 are more than ϵ apart, we conclude that these two ME grammars $G_{\mathbf{w}_{k_1}}^{\text{ME}}$ and $G_{\mathbf{w}_{k_2}}^{\text{ME}}$ are not ϵ -identical because they assign to the mapping (x_{k_2}, y) two probabilities that differ by more than ϵ . In conclusion, the ME typology $\mathfrak{T}^{\text{ME}}(\mathfrak{D}, \mathbf{C})$ corresponding to the phonological domain \mathfrak{D} in (15) and the constraint set \mathbf{C} in (16) is ϵ -infinite because it contains an infinite sequence

of ME grammars $G_{w_1}^{\text{ME}}, G_{w_k}^{\text{ME}}, \dots, G_{w_k}^{\text{ME}}, \dots$ which are pair-wise ϵ -different.

5.3 Generalization of result 1 to other distances

The distance D_∞ in (3) is obviously never larger than the distance D_1 in (4), as stated in (20a). Furthermore, we recall (see for instance Cover and Thomas 1991, page 300 and Tsybakov 2009, lemma 2.5, page 88) that the distance D_1 is never larger than twice the square root of the KL divergence D_{KL} in (5), as stated by Pinsker's inequality (20b). Finally, we recall (see for instance Tsybakov 2009, lemma 2.7, page 90) that the KL divergence D_{KL} is never larger than the χ^2 divergence D_{χ^2} in (6), yielding the inequality (20c).

$$\begin{aligned} D_\infty(G_1, G_2) &\stackrel{(a)}{\leq} D_1(G_1, G_2) \\ &\stackrel{(b)}{\leq} 2\sqrt{D_{\text{KL}}(G_1, G_2)} \\ &\stackrel{(c)}{\leq} 2\sqrt{D_{\chi^2}(G_1, G_2)} \end{aligned} \quad (20)$$

It follows from these inequalities (20) that, if a probabilistic typology is ϵ -infinite relative to the distance D_∞ , then it is also ϵ -infinite relative to the distance D_1 as well as relative to the divergences D_{KL} and D_{χ^2} . Since the ME typology $\mathfrak{T}^{\text{ME}}(\mathcal{D}, \mathcal{C})$ constructed in appendix 5.2 is ϵ -infinite relative to D_∞ , it is also ϵ -infinite relative to D_1 , D_{KL} , and D_{χ^2} . In other words, the result proved in appendix 5.2 is robust because it does not depend on how we measure the difference between probabilistic grammars.

5.4 A lemma for the proof of result 2

Lemma 2 Consider $k - 1$ vectors $\mathbf{d}_1, \dots, \mathbf{d}_{k-1}$ with integral components (without restrictions on their sign) and a vector \mathbf{w}_k with positive rational components such that $\mathbf{w}_k \cdot \mathbf{d}_1 > 0, \dots, \mathbf{w}_k \cdot \mathbf{d}_{k-1} > 0$. There exist a vector \mathbf{d}_k with integral components (without restrictions on their sign) and a vector \mathbf{w}_{k+1} with positive rational components such that $\mathbf{w}_{k+1} \cdot \mathbf{d}_1 > 0, \dots, \mathbf{w}_{k+1} \cdot \mathbf{d}_{k-1} > 0$ and furthermore $\mathbf{w}_{k+1} \cdot \mathbf{d}_k > 0$ while $\mathbf{w}_k \cdot \mathbf{d}_k < 0$. \square

This lemma admits the following geometric interpretation. We start from some vectors $\mathbf{d}_1, \dots, \mathbf{d}_{k-1}$, represented as blue dots in figure 7. They all sit in the interior of some half-space, represented as the blue region in figure 7a. We can always slightly tilt the surface that defines this half-space in such a way that the new half-space, represented as the

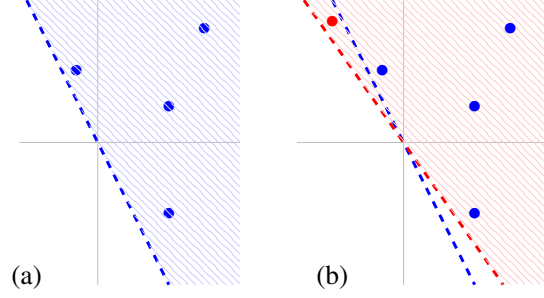


Figure 7

red region in figure 7b, satisfies the following two conditions. First, the original vectors $\mathbf{d}_1, \dots, \mathbf{d}_k$ sit in the interior of the tilted half-space as well. Second, we have made room for some new vector \mathbf{d}_{k+1} , represented by the red dot in figure 7b, that sits in the interior of the tilted red half-space but not of the original blue half-space.

To establish the lemma, we observe that, since the strict inequality $\mathbf{w}_k \cdot \mathbf{d}_h > 0$ holds for every $h = 1, \dots, k - 1$, there exists a positive rational constant $\epsilon > 0$ such that $\mathbf{w}_k \cdot \mathbf{d}_h \geq \epsilon$ for every $h = 1, \dots, k - 1$. Since the vector \mathbf{w}_k has rational components, there exists a vector \mathbf{v} with rational components orthogonal to \mathbf{w}_k , namely such that $\mathbf{v} \cdot \mathbf{w}_k = 0$. Let $M_1 > 0$ be the smallest common multiple of the denominators of the components of \mathbf{v} , whereby $M_1\mathbf{v}$ has integral components. Let $M_2 > 0$ be the smallest common multiple of the denominators of the components of \mathbf{w}_k , whereby $M_2\mathbf{w}_k$ has positive integral components. We choose a positive rational constant $\alpha > 0$ and a positive integer ℓ as in (21).

$$\begin{aligned} \alpha &= \begin{cases} 1 & \text{if } \beta \geq 0 \\ -\frac{\epsilon}{2\beta} & \text{if } \beta < 0 \end{cases} \quad \text{with } \beta = \min_{h=1}^{k-1} \mathbf{v} \cdot \mathbf{d}_h \\ \ell &\geq \frac{M_2 \|\mathbf{w}_k\|^2}{\alpha M_1 \|\mathbf{v}\|^2} \end{aligned} \quad (21)$$

We define the vector \mathbf{w}_{k+1} with positive rational components and the vector \mathbf{d}_k with integral components as in (22).

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha \mathbf{v} \quad \mathbf{d}_k = \ell M_1 \mathbf{v} - M_2 \mathbf{w}_k \quad (22)$$

These positions satisfy the inequalities (23) and (24) as well as the inequality (25) for $h =$

1, \dots, k-1, completing the proof of the lemma.

$$\begin{aligned} \mathbf{w}_k \cdot \mathbf{d}_k &= \mathbf{w}_k \cdot (\ell M_1 \mathbf{v} - M_2 \mathbf{w}_k) \quad (23) \\ &= -M_2 \|\mathbf{w}_k\|^2 < 0 \end{aligned}$$

$$\begin{aligned} \mathbf{w}_{k+1} \cdot \mathbf{d}_k &= (\mathbf{w}_k + \alpha \mathbf{v}) \cdot (\ell M_1 \mathbf{v} - M_2 \mathbf{w}_k) \quad (24) \\ &= \alpha \ell M_1 \|\mathbf{v}\|^2 - M_2 \|\mathbf{w}_k\|^2 > 0 \end{aligned}$$

$$\begin{aligned} \mathbf{w}_{k+1} \cdot \mathbf{d}_h &= (\mathbf{w}_k + \alpha \mathbf{v}) \cdot \mathbf{d}_h \quad (25) \\ &= \mathbf{w}_k \cdot \mathbf{d}_h + \alpha \mathbf{v} \cdot \mathbf{d}_h \geq \epsilon + \alpha \beta > 0 \end{aligned}$$

5.5 Proof of result 2

We start with an arbitrary vector \mathbf{w}_1 with positive rational components. By applying lemma 2 with $k = 1$ to this vector \mathbf{w}_1 , we conclude that there exist a vector \mathbf{d}_1 with integral components and a vector \mathbf{w}_2 with positive rational components that validate the red inequalities in the first step of the reasoning in figure 8. By applying again lemma 2 with $k = 2$ to the vectors \mathbf{d}_1 and \mathbf{w}_2 in the bottom line of this first step, we conclude that there exist a vector \mathbf{d}_2 with integral components and a vector \mathbf{w}_3 with positive rational components that validate the red inequalities in the second step of the reasoning in figure 8. By applying once again lemma 2 with $k = 3$ to the vectors $\mathbf{d}_1, \mathbf{d}_2$ and \mathbf{w}_3 in the bottom line of this second step, we conclude that there exist a vector \mathbf{d}_3 with integral components and a vector \mathbf{w}_4 with positive rational components that validate the red inequalities in the third step of the reasoning in figure 8. And so on and so forth.

In conclusion, we have established the existence of a sequence of vectors $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k, \dots$ with positive rational components and a sequence of vectors $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k, \dots$ with integral components that satisfy the k inequalities in (26) for every index $k = 1, 2, \dots$

$$\begin{aligned} \mathbf{w}_k \cdot \mathbf{d}_1 &> 0 & \mathbf{w}_k \cdot \mathbf{d}_k &< 0 \quad (26) \\ &\vdots & & \\ \mathbf{w}_k \cdot \mathbf{d}_{k-1} &> 0 \end{aligned}$$

To construct the desired counterexample, we consider the infinite phonological domain \mathfrak{D} described in (27). For every index $k = 1, 2, \dots$, the base set $B_{\mathfrak{D}}$ of this phonological domain contains the two underlying forms x_k and \hat{x}_k . Their candidate sets consist of only two surface forms, namely y, z and \hat{y}, \hat{z} , respectively. For concreteness, we refer to y and \hat{y} as the **winner** candidate and to z and \hat{z} as the **loser** candidates.

$$B_{\mathfrak{D}} = \left\{ \begin{array}{l} x_1 x_2 \dots x_k \dots \\ \hat{x}_1 \hat{x}_2 \dots \hat{x}_k \dots \end{array} \right\} \quad \begin{array}{l} \mathfrak{D}(x_k) = \{y, z\} \\ \mathfrak{D}(\hat{x}_k) = \{\hat{y}, \hat{z}\} \end{array} \quad (27)$$

Furthermore, we define the constraint set \mathbf{C} in such a way that the identity (28) holds for every index $k = 1, 2, \dots$. The first difference $\mathbf{C}(\hat{x}_k, \hat{z}) - \mathbf{C}(\hat{x}_k, \hat{y})$ on the righthand side compares the constraint violations of the loser and winner candidates \hat{z} and \hat{y} of the underlying form \hat{x}_k . The second difference $\mathbf{C}(x_k, z) - \mathbf{C}(x_k, y)$ compares the constraint violations of the loser and winner candidates z and y of the underlying form x_k that bears the same index k . The identity (28) says that the difference between these two differences must be equal to the vector \mathbf{d}_k in (26).

$$\mathbf{d}_k = \begin{pmatrix} \mathbf{C}(\hat{x}_k, \hat{z}) - \mathbf{C}(\hat{x}_k, \hat{y}) \\ -(\mathbf{C}(x_k, z) - \mathbf{C}(x_k, y)) \end{pmatrix} \quad (28)$$

This position (28) makes sense because the vector \mathbf{d}_k has integral components that can therefore be interpreted as differences between integral numbers of constraint violations. Furthermore, despite the fact that the components of this vector \mathbf{d}_k can be positive or negative, the identity (28) can always be satisfied by choosing constraint violation vectors such that $\mathbf{C}(\hat{x}_k, \hat{z}) \geq \mathbf{C}(\hat{x}_k, \hat{y})$ and $\mathbf{C}(x_k, z) \geq \mathbf{C}(x_k, y)$. This means that every constraint in the constraint set \mathbf{C} assigns less violations to the winner mapping (x_k, y) than to the loser mapping (x_k, z) ; analogously, it assigns less violations to the winner mapping (\hat{x}_k, \hat{y}) than to the loser mapping (\hat{x}_k, \hat{z}) . Equivalently, the winner mappings (x_k, y) and (\hat{x}_k, \hat{y}) always beat in HG the loser mappings (x_k, z) and (\hat{x}_k, \hat{z}) respectively, no matter the choice of the non-negative constraint weights. The HG typology $\mathfrak{T}^{\text{HG}}(\mathfrak{D}, \mathbf{C})$ corresponding to the phonological domain \mathfrak{D} in (27) and the constraint set \mathbf{C} in (28) therefore consists of a single HG grammar, namely the grammar that maps all the underlying forms x_k and \hat{x}_k to the candidates y and \hat{y} , respectively.

We now switch from categorical HG to probabilistic ME. We focus on the ME grammar $G_{\mathbf{w}_k}^{\text{ME}}$ corresponding to the weight vector \mathbf{w}_k in (26). We want to compare the probabilities it assigns to the two mappings (x_h, y) versus (\hat{x}_h, \hat{y}) with $h = 1, \dots, k-1$ as well as to the two mappings (x_k, y) versus (\hat{x}_k, \hat{y}) . As explained below, the inequalities $\mathbf{w}_k \cdot \mathbf{d}_h > 0$ with $h = 1, \dots, k-1$ on the lefthand side of (26) ensure that the ME grammar $G_{\mathbf{w}_k}^{\text{ME}}$ assigns less probability to the mapping (x_h, y) than to the mapping (\hat{x}_h, \hat{y}) for every $h = 1, \dots, k-1$, as stated in (29). Analogously, the $\mathbf{w}_k \cdot \mathbf{d}_k < 0$ on the righthand side of (26) ensures

$$\begin{array}{ccc}
\mathbf{w}_1 \implies & \begin{array}{l} \mathbf{w}_1 \cdot \mathbf{d}_1 < 0 \\ \mathbf{w}_2 \cdot \mathbf{d}_1 > 0 \end{array} & \implies \begin{array}{l} \mathbf{w}_1 \cdot \mathbf{d}_1 < 0 \\ \mathbf{w}_2 \cdot \mathbf{d}_1 > 0 \\ \mathbf{w}_3 \cdot \mathbf{d}_1 > 0 \end{array} \\
\text{first step} & & \text{second step} \\
& & \implies \begin{array}{l} \mathbf{w}_1 \cdot \mathbf{d}_1 < 0 \\ \mathbf{w}_2 \cdot \mathbf{d}_1 > 0 \\ \mathbf{w}_3 \cdot \mathbf{d}_1 > 0 \\ \mathbf{w}_4 \cdot \mathbf{d}_1 > 0 \end{array} \\
& & \text{third step}
\end{array}$$

Figure 8

that the ME grammar $G_{\mathbf{w}_k}^{\text{ME}}$ assigns more probability to the mapping (x_k, y) than to the mapping (\hat{x}_k, \hat{y}) , as stated in (30).

$$G_{\mathbf{w}_k}^{\text{ME}}(y | x_h) < G_{\mathbf{w}_k}^{\text{ME}}(\hat{y} | \hat{x}_h) \quad (29)$$

$$G_{\mathbf{w}_k}^{\text{ME}}(y | x_k) > G_{\mathbf{w}_k}^{\text{ME}}(\hat{y} | \hat{x}_k) \quad (30)$$

Indeed, the reasoning in (31) establishes the inequality (29) and an analogous reasoning establishes the inequality (30). Step (31a) holds by unpacking the ME probability as in steps (19a)-(19b) above. And tep (31b) holds because of the definition (28) of the vector \mathbf{d}_h . The condition $\mathbf{w}_k \cdot \mathbf{d}_h > 0$ arrived at is ensured by the choice of the vectors \mathbf{w}_k and \mathbf{d}_h in (26).

$$\begin{aligned}
G_{\mathbf{w}_k}^{\text{ME}}(y | x_h) &< G_{\mathbf{w}_k}^{\text{ME}}(\hat{y} | \hat{x}_h) & (31) \\
\stackrel{(a)}{\iff} & \frac{1}{1 + e^{-\mathbf{w}_k \cdot (\mathbf{C}(x_k, z) - \mathbf{C}(x_k, y))}} < \\
& < \frac{1}{1 + e^{-\mathbf{w}_k \cdot (\mathbf{C}(\hat{x}_k, \hat{z}) - \mathbf{C}(\hat{x}_k, \hat{y}))}} \\
\iff & e^{-\mathbf{w}_k \cdot (\mathbf{C}(x_k, z) - \mathbf{C}(x_k, y))} \\
& > e^{-\mathbf{w}_k \cdot (\mathbf{C}(\hat{x}_k, \hat{z}) - \mathbf{C}(\hat{x}_k, \hat{y}))} \\
\iff & \mathbf{w}_k \cdot (\mathbf{C}(x_k, z) - \mathbf{C}(x_k, y)) \\
& < \mathbf{w}_k \cdot (\mathbf{C}(\hat{x}_k, \hat{z}) - \mathbf{C}(\hat{x}_k, \hat{y})) \\
\stackrel{(b)}{\iff} & \mathbf{w}_k \cdot \mathbf{d}_h > 0
\end{aligned}$$

To complete the proof of result 2, we now consider the two ME grammars $G_{\mathbf{w}_{k_1}}^{\text{ME}}$ and $G_{\mathbf{w}_{k_2}}^{\text{ME}}$ corresponding to two weight vectors \mathbf{w}_{k_1} and \mathbf{w}_{k_2} with $k_1 > k_2$. By (29), the ME grammar $G_{\mathbf{w}_{k_1}}^{\text{ME}}$ assigns less probability to the mapping (x_{k_2}, y) than to the mapping (\hat{x}_{k_2}, \hat{y}) , namely $G_{\mathbf{w}_{k_1}}^{\text{ME}}(y | x_{k_2}) < G_{\mathbf{w}_{k_1}}^{\text{ME}}(\hat{y} | \hat{x}_{k_2})$. By (30), the ME grammar $G_{\mathbf{w}_{k_2}}^{\text{ME}}$ instead assigns more probability to the mapping (x_{k_2}, y) than to the mapping (\hat{x}_{k_2}, \hat{y}) , namely $G_{\mathbf{w}_{k_2}}^{\text{ME}}(y | x_{k_2}) > G_{\mathbf{w}_{k_2}}^{\text{ME}}(\hat{y} | \hat{x}_{k_2})$. These probability inequalities say that these two ME grammars $G_{\mathbf{w}_{k_1}}^{\text{ME}}$ and $G_{\mathbf{w}_{k_2}}^{\text{ME}}$ are not order-identical because they order the two mappings (x_{k_2}, y) and (\hat{x}_{k_2}, \hat{y}) differently. In conclusion, the ME typology $\mathfrak{T}^{\text{ME}}(\mathcal{D}, \mathbf{C})$ corresponding to the phonological

domain \mathcal{D} in (27) and the constraint set \mathbf{C} in (28) is order-infinite because it contains an infinite sequence of ME grammars $G_{\mathbf{w}_1}^{\text{ME}}, G_{\mathbf{w}_2}^{\text{ME}}, \dots, G_{\mathbf{w}_k}^{\text{ME}}, \dots$ which are pair-wise order-different.

Generating Feature Vectors from Phonetic Transcriptions in Cross-Linguistic Data Formats

Arne Rubehn¹, Jessica Nieder¹, Robert Forkel² and Johann-Mattis List¹

¹Chair for Multilingual Computational Linguistics, University of Passau
Passau, Germany

²DLCE, MPI-EVA
Leipzig, Germany

Abstract

When comparing speech sounds across languages, scholars often make use of feature representations of individual sounds in order to determine fine-grained sound similarities. Although binary feature systems for large numbers of speech sounds have been proposed, large-scale computational applications often face the challenges that the proposed feature systems – even if they list features for several thousand sounds – only cover a smaller part of the numerous speech sounds reflected in actual cross-linguistic data. In order to address the problem of missing data for attested speech sounds, we propose a new approach that can create binary feature vectors dynamically for all sounds that can be represented in the standardized version of the International Phonetic Alphabet proposed by the Cross-Linguistic Transcription Systems (CLTS) reference catalog. Since CLTS is actively used in large data collections, covering more than 2,000 distinct language varieties, our procedure for the generation of binary feature vectors provides immediate access to a very large collection of multilingual wordlists. Testing our feature system in different ways on different datasets proves that the system is not only useful to provide a straightforward means to compare the similarity of speech sounds, but also illustrates its potential to be used in future cross-linguistic machine learning applications.

1 Introduction

The past two decades have seen a drastic increase in standardized datasets in historical linguistics and linguistic typology which are available in both human- and machine-readable form (Dellert et al., 2020; Skirgård et al., 2023; Wichmann et al., 2013). With Lexibank (<https://lexibank.clld.org>, List et al. 2022), a large collection of comparative wordlists has been compiled in which word forms from various independently published datasets are standardized along three dimensions,

including (1) the languages in which they occur, (2) the concepts which they express, and (3) the sounds that constitute them. Lexibank is a result of the more general Cross-Linguistic Data Formats initiative (CLDF, <https://cldf.clld.org>, Forkel et al. 2018), which aims to unify several kinds of cross-linguistic data (wordlists, typological datasets, interlinearglossed texts) by proposing an exchange format along with guidelines and recommendations for standardization.

Sounds in Lexibank are represented in a unified transcription system, proposed as part of the Cross-Linguistic Transcription Systems reference catalogue (CLTS, <https://clts.clld.org>, List et al. 2024) that can be considered a large standardized subset of the International Phonetic Alphabet (IPA, IPA 1999). The CLTS system tries to handle as much of the variation observed in phonetic transcriptions as possible, using a dynamic method that parses phonetic transcriptions in a given transcription system and derives features from individual symbol combinations (Anderson et al., 2018). The feature system underlying these transcriptions has been designed in a pragmatic way that would allow to capture as much of the graphical variation in using the IPA (and other transcription systems) as possible (Anderson et al., 2023). As a result, the system is powerful in parsing phonetic transcriptions – specifically those represented in IPA – but it is not particularly useful to compare speech sounds with respect to their *similarity* (be it acoustic or articulatory or a combination of both).

Thus, while the CLTS system does its job in helping to standardize phonetic transcriptions in an unprecedented way, as witnessed by the Lexibank collection (and numerous additional CLDF wordlists that have been published in the past years), it falls short in providing a reliable means to compare individual sounds for their similarity.

In this study, we present a very straightforward approach to convert the CLTS feature system to a

vector representation. This approach takes CLTS feature bundles as input and converts them into binary feature vectors that can be used for various downstream tasks in computational phonology, computational historical linguistics, and computational linguistic typology.

2 Background

Modeling speech sounds as bundles of distinctive features can be seen as the most typical and straightforward way in phonology and comparative linguistics to compare the similarity of speech sounds. It is therefore not surprising, that phonological features play a crucial role in a number of different approaches, ranging from historical language comparison (Kondrak, 2000) over dialectology (Nerbonne and Heeringa, 1997; Hoppenbrouwers and Hoppenbrouwers, 2001) and phonological rule induction (Gildea and Jurafsky, 1996) to child language acquisition (Somers, 1998). Representing sounds with the help of features can also enhance the performance of automatic speech recognition (Metze, 2007) and transliteration (Tao et al., 2006; Yoon et al., 2007), as well as automatic phonetic transcriptions from text and named entity recognition (Mortensen et al., 2016). Related studies have additionally demonstrated that meaningful phone(me) embeddings can be learned from distributional properties (Silfverberg et al., 2018; Sofroniev and Çöltekin, 2018).

Beyond these mostly implicit uses of phonological features, there exist several frameworks with the explicit purpose of modeling the interactions between sounds and their respective features. Computational tools for analyzing phoneme inventories in terms of phonological features and natural classes are available in the form of web applications (Steel and Jurgec, 2017) or downloadable programs (van Vugt and Hayes, 2021). However, these tools by default cover rather small inventories of fairly common sounds and are often even only designed for individual languages.

There is a small number of datasets that map a large number of sounds to a feature representation, aiming to cover a substantially large amount of speech sounds in order to be applicable in cross-linguistic studies. PanPhon (Mortensen et al., 2016) defines feature representations for approximately 5,000 sounds, the similar but smaller framework DistFeat (Tresoldi, 2020) spans roughly 500 sounds. Phonotacticon (Joo and Hsu, 2023), a ty-

pological resource for phonotactics, extends PanPhon to around 20,000 distinct speech sounds. PHOIBLE (Moran and McCloy, 2019), a database covering various phoneme inventories, is equipped with feature definitions as well, covering all 3,000 distinct sounds attested in the data in its latest version. Finally, the Python package ipasymbols (Hamster, 2022) is designed to query IPA symbols by their articulatory properties, but is not equipped with phonological features and is currently (v.0.1.0) limited to only 179 sounds.

While all feature collections are much larger than the earlier feature collections that phonologists proposed for individual languages, reflecting the trend towards cross-linguistic approaches that allow for a comparison across multiple languages, all feature collections are *fixed sets of sounds*, lacking a dynamic component. This limits their potential when applying them to newly compiled datasets, since whenever a sound in a given dataset is not attested in the feature systems, users would have to add it or to label it as missing data.

While this may seem to reflect a minor problem, it has grown into a major obstacle for many concrete applications in computational comparative linguistics, since practical experience in working with concrete language data clearly shows that meeting unobserved sounds when turning to new datasets is rather the rule than the exception (see the observation in Moran 2012, that the overall number of distinct speech sounds seems to increase almost constantly, albeit slowly, when new data are added to the sample). One way to avoid the problem of observing missing sounds is to arbitrarily extend mappings from IPA transcriptions of speech sounds to feature mappings in a systematic way, as exemplified by the extended system proposed by the Phonotacticon, with 20,000 distinct sounds, of which only a couple of hundred sounds occur in the final database.

An alternative, more robust approach, specifically important for data standardized in CLDF, would take the pragmatically oriented non-binary features provided by the CLTS system as a starting point and convert them to a binary vector representation.

3 Materials and Methods

3.1 Materials

The starting point of our approach is the CLTS reference catalogue, which links feature descrip-

tions of speech sounds in the style of the IPA to different transcription systems and datasets. While the CLTS website presents a list of about 8,000 distinct speech sounds that are linked to various datasets, including PHOIBLE and PanPhon, the system that generates the website is dynamic, with only a couple of hundred base sounds being defined explicitly. The rest of the sounds is generated from sound transcriptions mainly provided in the International Phonetic Alphabet. The dynamic system underlying the CLTS reference catalogue can be accessed with the help of a Python API (<https://pypi.org/projects/pyclts>, List et al. 2020, see Anderson et al. 2018 for the details regarding the algorithm used by the API). As a result, IPA strings that are not directly represented in the system can be processed, as long as they conform to IPA standards (broadly defined by CLTS). The CLTS system parses sounds in two ways, taking a phonetic transcription (typically provided in IPA) as starting point, or starting from the typical name of a speech sound, as they are also defined by the IPA. For example, [p] would be described as the ‘voiceless bilabial stop consonant’, yielding the descriptive feature set (‘voiceless’, ‘bilabial’, ‘stop’, ‘consonant’). For the conversion of sound transcriptions accepted by CLTS to binary feature vectors, we use the feature bundle representation rather than the phonetic transcription as our starting point.

In order to evaluate the usefulness of the binary feature vectors derived from CLTS, we use the Lexibank database, since it provides a large collection of wordlists that conform to the standard defined by CLTS.

As of version 1.0 (List et al., 2023) Lexibank is available in an aggregated form in which all datasets that are sufficiently standardized – with all sounds being interpretable by the dynamic CLTS system – are assembled in a single CLDF dataset that can be parsed and processed in various ways, including SQLite (see List and Shcherbakova 2023) or Python (using the CL Toolkit package, see List and Forkel 2021, <https://pypi.org/project/cltoolkit>). In this form, Lexibank covers wordlists of at least 80 distinct words for about 2,000 distinct language varieties.

3.2 Methods

3.2.1 Feature System

We define a classical feature space of 39 binary phonological features that can be present (1) or

absent (-1), or non-applicable (0). Strictly speaking, we therefore employ a notion of ternary, rather than binary features, since there are three instead of two possible values. However, this is merely an explicit formalization of the way that binary features are usually treated in phonology: Not all features can apply to all kinds of sounds. It is therefore necessary to distinguish absent from non-applicable features by assigning them different numerical values. To illustrate this, consider the feature [\pm strident] which only applies to fricatives and affricates (Zsiga, 2013) – it is worthwhile to distinguish non-strident fricatives (which *could* be strident) from other sounds that do not have this feature at all. This notion of applicability is frequently found in the literature, and formally makes these systems ternary rather than binary. Keeping this in mind, we will still refer to this feature system as binary, given that it is the commonly used term to describe this kind of feature systems.

The majority of the features we define constitutes a fairly well established standard inventory, where we strictly follow the definitions by Zsiga (2013). Nevertheless, to be able to cover a comprehensive range of sounds, some additions to the feature inventory were required. We incorporate three additional features [velaric], [hitone], and [hireg] from Mortensen et al. (2016) to handle clicks and tones. Clicks are assigned [+velaric] on top of their other features that are derived from their analogous pulmonic stops. The tonal features [hireg] and [hitone] refer to the broader register, and the more narrow tone quality within the register – both the high tone [⁵] and the mid-high tone [⁴] belong to the high register and are therefore [+hireg], and within that register, [⁵] is the higher tone and is therefore [+hitone] (whereas [⁴] is [-hitone] analogously). Since we do not want to assign the feature value of 0 (non-applicable) to tonal features in tones, however, these two features only yield 4 possible combinations, insufficient for encoding the canonical 5 tones. We therefore introduce the supplementary feature [loreg] as a logical counterpart to [hireg], covering the low and the mid-low tones.

Furthermore, we employ three additional features to represent complex tones. Tonal features in CLTS are based on a rather schematic representation of Chao’s numeral coding of complex tones in Chinese dialects (Chao, 1930[2006]). Thus, the tone [²¹⁴] is labeled as “contour from-mid-low via-low to-mid-high tone” in the CLTS name

space, with the feature value "from-mid-low" (from the feature [start]) – representing the number 2, "via-low" (from the feature [middle]) representing 1, and "to-mid-high" (from the feature [end]) representing 4 directly, while the feature value "contour" (from the feature [contour]) adds additional information that tells us that we are dealing with a contour tone. In the same way, CLTS assigns the values "rising" and "falling" to tones like [15] and [51] respectively. In the vector representation, the contour of a tone is directly translated to the features [contour, rising, falling] based on their respective CLTS feature. Contour tones with both rising and falling parts additionally receive [+rising] or [+falling], indicating the interval between the initial and the middle segment. All complex tones inherit the features [hireg, hitone, loreg] from their initial segment. The example tone [214] is therefore represented as [-hireg, +hitone, +loreg, +contour, -rising, +falling].

Finally, seven more features are defined for representing diphthongs, namely [backshift, frontshift, opening, closing, centering, longdistance, seconddounded]. These features are used to model the diphthong’s trajectory across the vowel space (for a more detailed description see Rubehn, 2022, 41-43). Additionally, each diphthong is assigned the simple vowel features of its initial segment.

Due to its flexibility, the present system is highly customizable and can be used to generate different feature systems as well. Users can easily define their own feature inventories and mappings, according to their individual needs. The workflow presented in the following section is not dependent on the specific feature system and definitions that we suggest here.

3.2.2 Workflow

Our system generates binary feature vectors for any sound based on its feature set assigned by CLTS. Again, consider the example [p]: The method does not depart from the string "p", but from the feature set ('voiceless', 'bilabial', 'stop', 'consonant') that can easily be obtained from CLTS. The general workflow for generating binary feature vectors is outlined in Figure 1.

Underlying our system is a simple dictionary structure which maps triplets of CLTS feature values, CLTS feature domains, and binary feature representations onto each other. The feature value

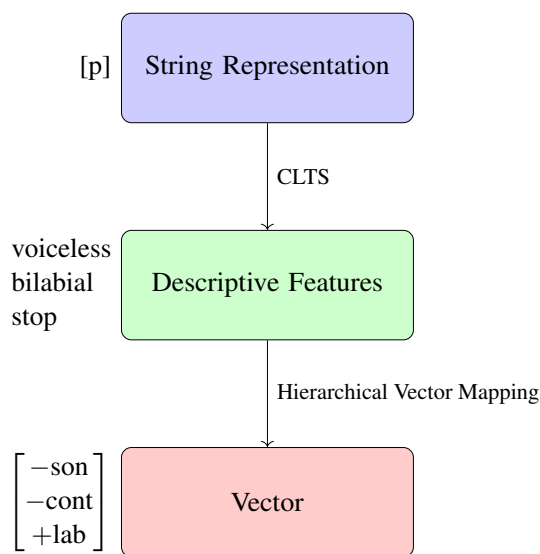


Figure 1: Workflow of vector creation.

‘stop’ for example would be linked to the domain ‘manner’ (of articulation in consonants) and to the binary feature representation [-son, -cont].

As a starting point, a zero vector (with the value 0 at every position) of the size of the defined feature inventory is instantiated, with every position of the vector corresponding to exactly one binary feature. This vector then is modified by subsequently processing the features in the CLTS feature set, with the corresponding binary features overwriting the current value in the vector.

A core principle for the successful modification of feature vectors is that we sort the CLTS features by a *hierarchy* of concreteness that determines the order in which those features are processed. This hierarchy states that the least specific features get processed first, and the most specific ones get processed last. In the concrete example of [p], that means that ‘consonant’, being the least specific feature, is processed first.

This notion of hierarchy is necessary for handling conflicting feature mappings, since we deliberately allow for values in the vector to be overwritten by features that are processed later. To exemplify this, consider the ‘devoiced voiced labiodental fricative’ [ɸ]: The descriptor ‘voiced’ maps to [+voice], whereas ‘devoiced’ naturally corresponds to [-voice]. However, since diacritics modify the base sound, they should take precedence over it, and the correct feature that should be assigned is [-voice]. This is ensured by processing the features according to the hierarchy, which

states that the modification ‘devoiced’ is more concrete and should therefore be applied *after* the regular phonation feature ‘voiced’, and can thus overwrite the previously assigned [+voice] with [-voice].

This notion of hierarchy also conveniently allows for the usage of *default values* that can define which features apply at which representation level. This is important since we distinguish between non-applicable (0) and absent (-1) features, as discussed in Section 3.2.1. We can therefore define which set of binary features always applies to a certain group of sounds by assigning a default value to ensure that applicable features have a non-zero value. For example, the feature [\pm lab(ial)] must be defined for all consonants, which is assured by mapping the CLTS feature ‘consonant’ to [-lab]. If the consonant is actually labial, this feature will be overwritten with [+lab], since the place of articulation is always applied after the sound type. So instead of exhaustively defining [-lab] for every non-labial place of articulation, we can just define it as a default value for the CLTS feature ‘consonant’ instead. This corresponds to the reading that every consonant is [-lab] (by default), unless specified otherwise.

The majority of sounds can be handled by this straightforward workflow of hierarchically mapping CLTS features to their binary feature representations. However, there are a few more complex cases that require an extra processing step. For example, the glottal stop [ʔ] has the binary feature [+cg] (‘constricted glottis’) – however, this feature neither corresponds to ‘glottal’, nor to ‘stop’. It is therefore the combination of both ‘glottal’ and ‘stop’ that triggers [+cg]. The system therefore uses a second dictionary that allows for the definition of joint feature mappings, where a binary feature definition is conditioned by a certain *combination* of CLTS features.

Complex sounds that can alternatively be analyzed as two segments – diphthongs and consonant clusters – pose a similar challenge. For these cases, CLTS provides the means of analyzing its individual constituents: The consonant cluster [kp] can be split into [k] and [p]. The system uses this function to generate separate feature vectors for the two individual sounds, which then are combined by assigning the union of positive features to the joint vector. The resulting feature vector for [kp] therefore contains all positive features that are attributed to either [k] or [p].

In a similar fashion, feature vectors for diphthongs are based on their initial segments. For example, [ai] inherits its monophthong vowel features from the feature definitions for [a]. The additional diphthong features, that indicate the trajectory of the diphthong, are assigned based on joint feature definitions: The combination of the CLTS features (‘from_open’, ‘to_near-close’) maps (among others) to the binary feature [+closing].

3.3 Implementation

The approach is implemented in the form of a Python package (soundvectors) that takes as input the canonical names consisting of feature values that CLTS generates dynamically for speech sounds in standard IPA transcription and can be applied in combination with CLTS and the pylts package, as well as with the linse package that offers non-generative access to a larger selection of speech sounds covered by CLTS (<https://pypi.org/project/linse>, List and Forkel 2024), but also independently of these packages, as long as the feature names follow the CLTS standards. The package along with the data on which it was tested is available from the supplementary material accompanying this study.

4 Evaluation

We test the usefulness of our proposed system by (1) investigating the vector similarities for common sounds by calculating cosine similarities and visualizing them with heatmaps, (2) employing techniques for dimensionality reduction to visualize the relationships between sounds, (3) mapping the CLTS sound inventory to binary vectors and analyzing the resulting equivalence classes, and (4) investigating the power of the system to distinguish speech sounds observed in phonetically transcribed wordlists.

4.1 Vector Similarities

To test how well our system analyzes a representative sample of common sounds, we take 25 most common consonants and the 20 most common vowels from Phoible 2.0 (Moran and McCloy, 2019) and use heatmaps to visualize the cosine similarities of their respective feature vectors (Figures 2 and 3). The heatmaps were generated with the Python library Seaborn (Waskom, 2021), with lighter colours representing higher similarities and darker colours representing lower similarity scores. The sounds are ordered by their primary

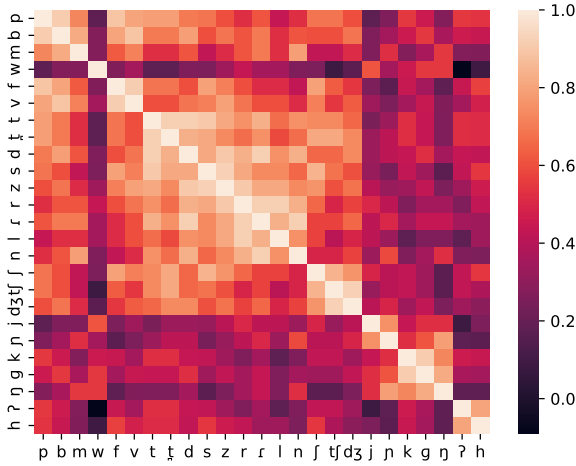


Figure 2: Cosine similarities between consonant vectors generated with our model.

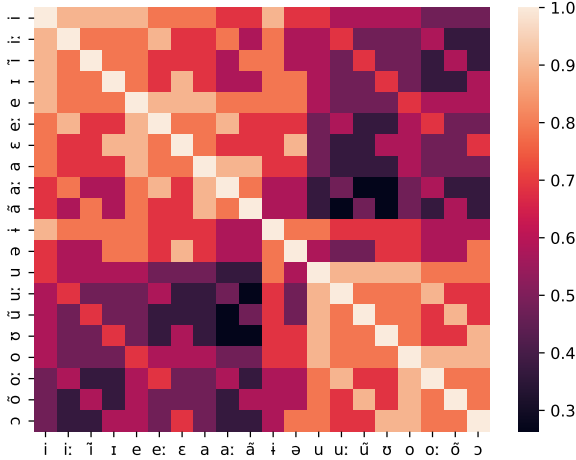


Figure 3: Cosine similarities between vowel vectors generated with our model.

place of articulation, starting from the front and moving to the back of the mouth. As can be observed in Figure 2, the manner of articulation and the phonation have a clear impact on the similarity of consonantal feature vectors: [p] is therefore much more similar to [k] than to [ŋ]. The glides [w] and [j] are strikingly dissimilar to the rest of the consonants, showing their well-known intermediate role in between consonants and vowels.

Figure 3 shows that the vowel space follows a strong division into two clusters which correspond to front and back vowels, with [a] being considered a front vowel according to the IPA nomenclature. This primary partition reflects the fact that typologically unmarked front vowels are unrounded, and back vowels are typically rounded. This naturally translates into a separate feature which drives front

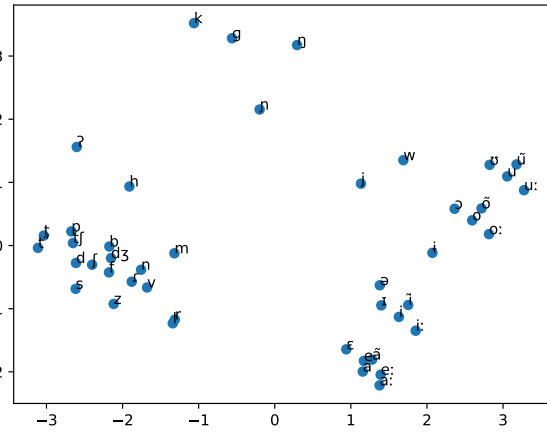


Figure 4: Two-dimensional reduction of feature vectors using PCA.

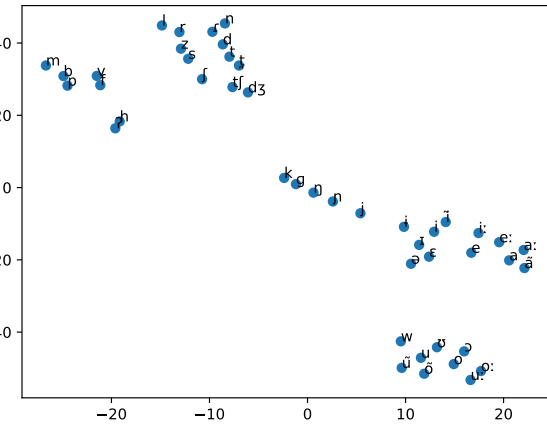


Figure 5: Two-dimensional reduction of feature vectors using t-SNE.

and back vowels further apart in terms of their vector similarity. Nonetheless, vowel pairs that share the same height such as [e] and [o] retain a fairly high degree of cosine similarity, indicating that our vectors are able to adequately reflect and apply vowel features to the used vowel data set.

4.2 Dimensionality Reduction

We employed two techniques for dimensionality reduction to project the phonological vector space onto a two-dimensional plane, aiming to gain a comprehensive understanding of the inherent structure of our vectors. The employed techniques are *principal component analysis* (PCA; Figure 4), known for its ability to reveal global linear structures, and *t-distributed stochastic neighbor embedding* (t-SNE; Figure 5; van der Maaten and Hinton 2008), known for capturing local nonlinear patterns in the data. Both dimensionality reductions were computed in Python using the SciKit-Learn pack-

age (Pedregosa et al., 2011) and visualized with Matplotlib (Hunter, 2007). By employing PCA and t-SNE at the same time, we sought to ensure a robust and detailed exploration of our vectors, leveraging the complementary strengths of both of these dimensionality reduction techniques (Anowar et al., 2021).

Figure 4 and Figure 5 visualize the results after dimensionality reduction. Both PCA and t-SNE reveal a consistent narrative, grouping similar sounds together. First and foremost, there is a clear distinction between consonants and vowels, with semi-vowels positioned either between the two clusters in PCA or in close proximity to the corresponding vowel cluster in t-SNE. Once again, this reaffirms the ability of our binary vectors to adequately distinguish between the sounds in a phonologically informed way.

Focusing on the vowel clusters in Figure 4 and Figure 5, it becomes evident once again that vowels are primarily divided into (unrounded) front and (rounded) back vowels, aligning with the well-established phonological classification of vowel sounds. Shifting attention to the consonant clusters in both panels, we once more observe that they are primarily grouped by their place of articulation. In both Figures, the velars [k,g,ŋ] form a rather isolated cluster, loosely associated with the palatal nasal [ɲ]. Both techniques also tend to isolate the glottal sounds [h,ʔ], with t-SNE placing this sound pair much closer to each other. A distinct picture emerges for the remaining consonants, the coronals and labials: While PCA seems to bundle all of them together into a single large cluster, t-SNE forms two distinct clusters based on their place of articulation, however retaining a certain proximity between the two. The t-SNE plot also exhibits a compelling parallelogram symmetry among quadruplets of stops and fricatives in their voiced and voiceless versions: The alveolars [t,d,s,z] form one such parallelogram in the two-dimensional projection; and a similar pattern can be observed for labials [p,b,f,v].

The observed similarities depicted in Figure 2 and Figure 3 as well as the patterns in the vowel and consonant clusters depicted in Figure 4 and Figure 5 align with established phonological classifications, providing a visual representation that echoes the theoretical descriptions and classes in phonological theory to a considerable extent. Our observations confirm the potential utility of our phonological feature vectors in computational models, suggesting that the vectors capture meaningful

distinctions and relationships inherent in the sounds of human languages.

4.3 Equivalence Classes

The current version of CLTS (v.2.1.0) provides a collection of 8,684 unique sounds that were observed in its source datasets. Employing our system, these 8,684 sounds map to 5,285 distinct feature vectors. The system is therefore capable of providing a unique representation for 60.9% of this large sound inventory, even though it contains a number of very narrow transcriptions, or aspects that we deliberately chose not to represent in the feature space, such as suprasegmental properties being attributed to a segment (for example, putting tones on vowels).

The largest two equivalence classes contain 18 segments respectively, which are all mid and open-mid vowels. The first class therefore contains sounds that are based on [ə] and [ɜ], including among others [ɔ̃, œ̃, ʒ̃]. All of these modifications are deliberately disregarded by our system: Tones are suprasegmental features that should not be represented as part of a segment, rhotics lack reliable phonetic correlates (Chabot, 2019), and specifying the relative tongue position is an overly narrow transcription style that does not carry distinctive information.

This illustrates the principle of economy, in that we only define as many features as strictly necessary to keep the individual features meaningful and avoid feature inflation. The distinctions that are lost by employing this procedure are extremely narrow in domain, and phonetically not meaningful, as we will argue in the following section.

4.4 Distinctiveness

We investigate the discriminative potential of our system by applying it to all sound inventories observed in phonetic transcriptions of lexical data in Lexibank (List et al., 2022, 2023). The aggregated dataset combines numerous datasets into one unified dataset, spanning over 2,905 language varieties in total. In Table 1, we report the metrics of how well the sound inventories of the languages in Lexibank 1.0 can be described by our system. We report the number of confused sounds per language, that is how many sounds in an inventory share their feature representation with another sound present in this inventory. Formally, this is the difference between the size of the sound inventory, and the number of unique feature representations corresponding

n confused sounds	n varieties	Portion
0	2,376	0.818
≤ 1	2,567	0.884
≤ 2	2,648	0.912
≤ 3	2,689	0.926
≤ 4	2,841	0.978

Table 1: Number of language varieties in Lexibank 1.0 with at most n confused sounds.

to the inventory.

The sound inventories of 2,376 varieties, amounting to 81.8% of all varieties in the dataset, can be represented with full distinctiveness by our system, meaning that every sound is mapped to a unique feature vector. With 2,841 (97.8%) varieties, the grand share of the dataset’s sound inventories can be represented with a maximum of four overlapping feature representations. These overlaps can usually be explained by narrow transcriptions, where the exact realization of the sound is predictable from the context, or can even hint at transcription errors or inconsistencies in the source data.

We investigate such sets of overlapping sounds in their context using concordance lines. This technique is frequently used in corpus linguistics for visualizing in which contexts a certain word appears. Usually, concordance lines are generated by aligning the highlighted target word to the center of a table, and placing the contexts to its left or right respectively (Hunston, 2022, 47).

annoy	e m f a ð a r
be annoyed	e m f a ð a r s e
be ill	e s t a r + e m f e r m o
fall ill	k a e r + e m f e r m o
illness	e m f e r m e ð a ð
sick, ill	e m f e r m o
December	d i θ j e m b r e
March	m a r θ o
May	m a j o
November	n o β j e m b r e
September	s e β t j e m b r e
Sunday	d o m i ŋ g o

Figure 6: Concordance line for Spanish transcriptions featuring [ŋ] or [m].

Figure 6 exemplifies the usefulness of concor-

dance lines to analyze the contexts in which sounds occur. Here, we investigate the instances of the bilabial and labio-dental nasal consonants [m] and [ŋ] in the Spanish data from the NorthEuraLex database (Dellert et al., 2020). Both sounds are represented by identical feature vector, but the data makes a distinction between them. Analyzing the relevant forms, however, shows that the presence of [ŋ] can be clearly predicted from the context, since it can only occur before labio-dental obstruents. This suggests that there is no actual distinction between these sounds, since the different surface forms can be explained by a fully predictable assimilation process. Similar cases of complementary distribution can be observed within the same dataset: In Nanai, [j] is only found preceding bilabial and alveolar consonants, while [i] occurs elsewhere; Estonian [k] is transcribed as [k̚] if preceded by [ŋ]; and Korean voiceless stops are unreleased in word-final position, leading to pairs of distinct transcriptions (e.g. [t] - [t̚]) with a distribution that is completely predictable by context.

5 Discussion

In this study, we introduced a new approach to turn the features for all sounds covered by the CLTS reference catalogue into numerical feature vectors. Given that CLTS not only underlies the Lexibank repository, which offers phonetically transcribed, standardized wordlists for more than 2,000 language varieties, but is also used in many additional applications that make use of the standards proposed by the CLDF initiative, this means that the binary feature vectors we propose are directly available for a very large number of language varieties.

To assess the effectiveness of our approach, we conducted a detailed analysis using cosine similarity and dimensionality reduction techniques. The resulting similarity patterns, evident in both PCA and t-SNE plots, align with established phonological classifications. Notably, the model distinguishes between vowels and consonants and groups similar sounds based on their place of articulation. Furthermore, we successfully mapped a substantial inventory of sounds from CLTS to their respective vectors, covering more than half of this extensive sound dataset with unique representations. Finally, we evaluated the distinctiveness of our vector representations by discerning speech sounds from lexical data in the Lexibank repository. Our system accurately represented a significant portion of the

data, ensuring full distinctiveness by uniquely mapping each sound to a feature vector.

In conclusion, our approach not only provides a practical solution to address general limitations of the pragmatic feature system underlying the CLTS reference catalogue but also offers a flexible approach for representing phonological features in computational linguistics. By converting CLTS feature bundles into binary feature vectors, the approach enables researchers to integrate phonological insights into various computational tasks, ranging from phonology and historical linguistics to linguistic typology.

For the field of cognitive language modeling, our feature system offers an enhanced, more precise phonological representation. Nieder and List (2024) utilized historical sound class representations in their language processing model (Linear Discriminative Learning, see Nieder and List, 2024) to explore mutual intelligibility among Germanic languages. Expanding such models with phonological vector representations instead, may offer new insights into how speech sounds influence meaning and vice versa, thereby guiding language processing and language learning.

For historical language comparison, feature representations can be used to dynamically extend fixed-size scoring matrices in computational tasks such as phonetic alignment (Kondrak, 2000; List, 2012) or phonological reconstruction (Bouchard-Côté et al., 2013; Jäger, 2019; Meloni et al., 2021). While state-of-the-art approaches to phonetic alignment typically deal with the problem of unseen sounds by resorting to sound class representations that represent sounds in phonetic transcriptions in small classes of similar sounds ranging from 10 to about 40 distinct sound classes in total (see List 2014 for details on sound class systems), feature vectors would offer a much more fine-grained representation of similarities and differences between individual sounds whose impacts on alignment quality have not been fully tested so far (an exception is the feature-based system by Kilani 2020, which requires, however, sound-feature mappings to be set up manually). For the still unsolved task of unsupervised phonological reconstruction (List, 2023), a common problem of those approaches that have been proposed so far is that they cannot propose sounds in ancestral languages that have not been attested in the descendant languages. Here, feature vectors might propose a way of handling the unknown, since the vector representation might well

propose feature combinations for ancestral sounds that are not observed in individual languages, thus creating unseen sounds from attested sounds. But further tests would be needed to explore the potential of feature vectors in phonological reconstruction.

In summary, we hope that feature vectors, as they have been introduced here, will prove useful in advancing computational approaches in linguistics and integrating linguistic insights into machine learning approaches.

Supplementary Materials

Data and code of this study are curated on GitHub (<https://github.com/cldf-clts/soundvectors>), the `soundvectors` package is also available via the Python package repository PyPi (<https://pypi.org/project/soundvectors>, Version 1.0).

Limitations

As discussed in Sections 4.3 and 4.4, we want our system to distinguish sounds that sound differently and avoid lumping them together. Our quantitative and qualitative analyses show that the system seems to be capable of maintaining a high degree of distinctiveness, however, it is not guaranteed that all phonemic contrasts in the world’s languages are represented truthfully. While our phonological feature vectors are a good approximation to spoken language, we want to point out that they cannot perfectly reflect phonetic similarity – some features are intuitively more meaningful than others, which is not explicitly represented in the vector space; and by extension, similar sounds might differ in a “disproportionately large number of features” (Kondrak, 2000). Heeringa (2004) shows that employing binary features directly as a cost function is not superior to plain edit distance to measure phonetic similarity in dialectal data.

These findings do not undermine the potential of feature systems in analyzing sounds, but rather show that feature vectors should be processed in some way and not be taken at face value. In fact, a number of studies have shown the usefulness of phonological feature vectors for processing sounds in machine learning approaches (e.g. Staib et al. 2020; Lux and Vu 2022). This emphasizes the need for a robust system that reliably generates feature vectors for all IPA segments.

Acknowledgements

The authors would like to thank Johannes Dellert for fruitful discussions on the choice of the feature inventory and several feature mappings, as well as two anonymous reviewers for their helpful comments and suggestions. This project was supported by the ERC Consolidator Grant ProduSemy (PI Johann-Mattis List, Grant No. 101044282, see <https://doi.org/10.3030/101044282>). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency (nor any other funding agencies involved). Neither the European Union nor the granting authority can be held responsible for them.

References

- Cormac Anderson, Tiago Tresoldi, Thiago Chacon, Anne-Maria Fehn, Mary Walworth, Robert Forkel, and Johann-Mattis List. 2018. *A cross-linguistic database of phonetic transcription systems*. *Yearbook of the Poznań Linguistic Meeting*, 4(1):21–53.
- Cormac Anderson, Tiago Tresoldi, Simon J. Greenhill, Robert Forkel, Russell D. Gray, and Johann-Mattis List. 2023. *Variation in phoneme inventories: quantifying the problem and improving comparability*. *Journal of Language Evolution*, 0(0):1–20.
- Farzana Anowar, Samira Sadaoui, and Bassant Selim. 2021. *Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE)*. *Computer Science Review*, 40:100378.
- Alexandre Bouchard-Côté, David Hall, Thomas L. Griffiths, and Dan Klein. 2013. *Automated reconstruction of ancient languages using probabilistic models of sound change*. *Proceedings of the National Academy of Sciences of the United States of America*, 110(11):4224–4229.
- Alex Chabot. 2019. *What’s wrong with being a rhotic?* *Glossa: a journal of general linguistics*, 4(1).
- Yuenren Chao. 1930[2006]. *A system of ‘tone letters’*. In Z.-j. Wu and X.-n Zhao, editors, *Linguistic Essays by Yuenren Chao*, pages 98–102. Shāngwù, Běijīng.
- Johannes Dellert, Thora Daneyko, Alla Münch, Alina Ladygina, Armin Buch, Natalie Clarius, Ilja Grigorjew, Mohamed Balabel, Hizniye Isabella Boga, Zalina Baysarova, et al. 2020. *NorthEuraLex: A wide-coverage lexical database of Northern Eurasia*. *Language resources and evaluation*, 54(1):273–301.
- Robert Forkel, Johann-Mattis List, Simon J. Greenhill, Christoph Rzymiski, Sebastian Bank, Michael Cysouw, Harald Hammarström, Martin Haspelmath, Gereon A. Kaiping, and Russell D. Gray. 2018. *Cross-Linguistic Data Formats, advancing data sharing and re-use in comparative linguistics*. *Scientific data*, 5(1):1–10.
- Daniel Gildea and Daniel Jurafsky. 1996. *Learning bias and phonological-rule induction*. *Computational Linguistics*, 22(4):497–530.
- Ulf A. Hamster. 2022. *Everybody likes short sentences - a data analysis for the text complexity DE challenge 2022*. In *Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text*, pages 10–14, Potsdam, Germany. Association for Computational Linguistics.
- Wilbert Heeringa. 2004. *Measuring dialect pronunciation differences using Levenshtein distance*. Ph.D. thesis, Rijksuniversiteit Groningen.
- Cor Hoppenbrouwers and Geer Hoppenbrouwers. 2001. *De indeling van de Nederlandse streektalen: Dialecten van 156 steden en dorpen geklasseerd volgens de FFM (feature frequentie methode)*. Koninklijke Van Gorcum, Assen.
- Susan Hunston. 2022. *Corpora in applied linguistics*, 2 edition. Cambridge University Press, Cambridge.
- John D. Hunter. 2007. *Matplotlib: A 2d graphics environment*. *Computing in Science & Engineering*, 9(3):90–95.
- IPA. 1999. *Handbook of the International Phonetic Association. A guide to the use of the international phonetic alphabet*. Cambridge University Press, Cambridge.
- Ian Joo and Yu-Yin Hsu. 2023. *Phonotacticon: A cross-linguistic phonotactic database*. Preprint, Research Square Platform LLC.
- Gerhard Jäger. 2019. *Computational historical linguistics*. *Theoretical Linguistics*, 45(3-4):151–182.
- Marwan Kilani. 2020. *FAAL: a feature-based aligning ALgorithm*. *Language Dynamics and Change*, 11(1):30–76.
- Grzegorz Kondrak. 2000. *A new algorithm for the alignment of phonetic sequences*. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.
- Johann-Mattis List. 2012. *SCA: Phonetic alignment based on sound classes*. In Marija Slavkovic and Dan Lassiter, editors, *New directions in logic, language, and computation*, pages 32–51. Springer, Berlin and Heidelberg.
- Johann-Mattis List. 2014. *Sequence comparison in historical linguistics*. Düsseldorf University Press, Düsseldorf.

- Johann-Mattis List. 2023. [Open problems in computational historical linguistics \[version 1; peer review: 3 approved, 2 approved with reservations\]](#). *Open Research Europe*, 3(201):1–22.
- Johann-Mattis List, Cormac Anderson, Tiago Tresoldi, and Robert Forkel. 2020. *PYCLTS. A Python library for the handling of phonetic transcription systems [Software Library, Version 3.0.0]*. Zenodo, Geneva.
- Johann-Mattis List, Cormac Anderson, Tiago Tresoldi, and Robert Forkel. 2024. *Cross-Linguistic Transcription Systems [Dataset, Version 2.3.0]*. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Johann-Mattis List and Robert Forkel. 2021. *CL Toolkit. A Python Library for the Processing of Cross-Linguistic Data [Software Library, Version 0.1.1]*. Zenodo, Geneva.
- Johann-Mattis List and Robert Forkel. 2024. [A new Python library for the manipulation and annotation of linguistics sequences](#). *Computer-Assisted Language Comparison in Practice*, 7(1):17–23.
- Johann-Mattis List, Robert Forkel, Simon J. Greenhill, Christoph Rzymiski, Johannes Englisch, and Russell D. Gray. 2022. [Lexibank, a public repository of standardized wordlists with computed phonological and lexical features](#). *Scientific Data*, 9(1):1–16.
- Johann-Mattis List, Robert Forkel, Simon J. Greenhill, Christoph Rzymiski, Johannes Englisch, and Russell D. Gray. 2023. *Lexibank [Database, Version 1.0]*. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Johann-Mattis List and Olena Shcherbakova. 2023. [Retrieving and Analyzing Taste Colexifications from Lexibank](#). *Computer-Assisted Language Comparison in Practice*, 6(2):73–86.
- Florian Lux and Thang Vu. 2022. [Language-agnostic meta-learning for low-resource text-to-speech with articulatory features](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6858–6868, Dublin, Ireland. Association for Computational Linguistics.
- Carlo Meloni, Shauli Ravfogel, and Yoav Goldberg. 2021. [Ab antiquo: Neural proto-language reconstruction](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4460–4473, Online. Association for Computational Linguistics.
- Florian Metze. 2007. [On using articulatory features for discriminative speaker adaptation](#). In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 117–120.
- Steven Moran. 2012. *Phonetics Information Base and Lexicon*. Phd, University of Washington.
- Steven Moran and Daniel McCloy, editors. 2019. *PHOIBLE 2.0*. Max Planck Institute for the Science of Human History, Jena.
- David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. [Panhon: A resource for mapping IPA segments to articulatory feature vectors](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484.
- John Nerbonne and Wilbert Heeringa. 1997. [Measuring dialect distance phonetically](#). In *Computational phonology: Third Meeting of the ACL Special Interest Group in Computational Phonology*.
- Jessica Nieder and Johann-Mattis List. 2024. [A computational model for the assessment of mutual intelligibility among closely related languages](#). In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, page 37–43, St. Julian’s, Malta. Association for Computational Linguistics.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research*, 12:2825–2830.
- Arne Rubehn. 2022. [A feature-based neural model of sound change informed by global lexicostatistical data](#). Master’s thesis, University of Tübingen.
- 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*, pages 136–144, Salt Lake City, Utah.
- Hedvig Skirgård, Hannah J. Haynie, Damián E. Blasi, Harald Hammarström, Jeremy Collins, Jay J. Latache, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Sam Passmore, Angela Chira, Luke Maurits, Russell Dinnage, Michael Dunn, Ger Reesink, Ruth Singer, Claire Bowerman, Patience Epps, Jane Hill, Outi Vesakoski, Martine Robbeets, Noor Karolin Abbas, Daniel Auer, Nancy A. Bakker, Giulia Barbos, Robert D. Borges, Swintha Danielsen, Luise Dorenbusch, Ella Dorn, John Elliott, Giada Falcone, Jana Fischer, Yustinus Ghanggo Ate, Hannah Gibson, Hans-Philipp Göbel, Jemima A. Goodall, Victoria Gruner, Andrew Harvey, Rebekah Hayes, Leonard Heer, Roberto E. Herrera Miranda, Nataliia Hübler, Biu Huntington-Rainey, Jessica K. Ivani, Marilen Johns, Erika Just, Eri Kashima, Carolina Kipf, Janina V. Klingenberg, Nikita König, Aikaterina Koti, Richard G. A. Kowalik, Olga Krasnoukhova, Nora L. M. Lindvall, Mandy Lorenzen, Hannah Lutzenberger, Tânia R. A. Martins, Celia Mata German, Suzanne van der Meer,

- Jaime Montoya Samamé, Michael Müller, Saliha Muradoglu, Kelsey Neely, Johanna Nickel, Miina Norvik, Cheryl Akinyi Oluoch, Jesse Peacock, India O. C. Pearey, Naomi Peck, Stephanie Petit, Sören Pieper, Mariana Poblete, Daniel Prestipino, Linda Raabe, Amna Raja, Janis Reimringer, Sydney C. Rey, Julia Rizaew, Eloisa Ruppert, Kim K. Salmon, Jill Sammet, Rhiannon Schembri, Lars Schlabbach, Frederick W. P. Schmidt, Amalia Skilton, Wikaliler Daniel Smith, Hilário de Sousa, Kristin Sverredal, Daniel Valle, Javier Vera, Judith Voß, Tim Witte, Henry Wu, Stephanie Yam, Jingting Ye, Maisie Yong, Tessa Yuditha, Roberto Zariquiey, Robert Forkel, Nicholas Evans, Stephen C. Levinson, Martin Haspelmath, Simon J. Greenhill, Quentin D. Atkinson, and Russell D. Gray. 2023. [Grambank reveals the importance of genealogical constraints on linguistic diversity and highlights the impact of language loss](#). *Science Advances*, 9(16).
- Pavel Sofroniev and Çağrı Çöltekin. 2018. [Phonetic vector representations for sound sequence alignment](#). In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 111–116, Brussels, Belgium.
- Harold Somers. 1998. [Similarity metrics for aligning children’s articulation data](#). In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1227–1232.
- Marlene Staib, Tian Huey Teh, Alexandra Torresquintero, Devang S. Ram Mohan, Lorenzo Foglianti, Raphael Lenain, and Jiameng Gao. 2020. [Phonological Features for 0-shot Multilingual Speech Synthesis](#). In *Proceedings of INTERSPEECH 2020*, Shanghai, China.
- George Steel and Peter Jurgec. 2017. [Featurize!: An online tool for natural classes and phonological features](#). University of Toronto, Toronto.
- Tao Tao, Su-Youn Yoon, Andrew Fister, Richard Sproat, and ChengXiang Zhai. 2006. [Unsupervised named entity transliteration using temporal and phonetic correlation](#). In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 250–257.
- Tiago Tresoldi. 2020. [A model of distinctive features for computer-assisted language comparison](#). *Computer-Assisted Language Comparison in Practice*, 3(6).
- Laurens van der Maaten and Geoffrey Hinton. 2008. [Visualizing data using t-SNE](#). *Journal of machine learning research*, 9(11).
- Floris van Vugt and Bruce Hayes. 2021. [Pheatures Spreadsheet: User’s Manual](#). UCLA, Los Angeles.
- Michael L. Waskom. 2021. [seaborn: statistical data visualization](#). *Journal of Open Source Software*, 6(60):3021.
- Søren Wichmann, André Müller, Annkathrin Wett, Viveka Velupillai, Julia Bischoffberger, Cecil H. Brown, Eric W. Holman, Sebastian Sauppe, Zarina Molochieva, Pamela Brown, Harald Hammarström, Oleg Belyaev, Johann-Mattis List, Dik Bakker, Dmitry Egorov, Matthias Urban, Robert Mailhammer, Agustina Carrizo, Matthew S. Dryer, Evgenia Korovina, David Beck, Helen Geyer, Pattie Epps, Anthony Grant, and Pilar Valenzuela. 2013. [The ASJP Database](#). Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Su-Youn Yoon, Kyoung-Young Kim, and Richard Sproat. 2007. [Multilingual transliteration using feature based phonetic method](#). In *Proceedings of the 45th Annual meeting of the association of computational linguistics*, pages 112–119.
- Elizabeth C. Zsiga. 2013. *The Sounds of Language: An Introduction to Phonetics and Phonology*, volume 7. John Wiley & Sons.

Computing Ellipsis Constructions: Comparing Classical NLP and LLM Approaches

Damir Cavar
Indiana University
dcavar@iu.edu

Zoran Tiganj
Indiana University
ztiganj@iu.edu

Ludovic Veta Mompelat
University of Miami
lvm861@miami.edu

Billy Dickson
Indiana University
dicksonb@iu.edu

Abstract

Although ellipsis constructions are highly frequent in common genres and discourse in all languages, State-of-the-art (SOTA) Natural Language Processing (NLP) technologies face significant challenges with such constructions. While the phenomenon as such is theoretically well-documented and understood, current technologies fail to provide adequate syntactic and semantic analyses due to many factors. One of those factors is insufficient cross-linguistic language resources covering ellipsis and ultimately serving the engineering of NLP solutions that more adequately provide correct analyses for ellipsis constructions. This article describes our effort to create a dataset that currently covers more than eighteen languages. We demonstrate how SOTA parsers based on a variety of syntactic frameworks fail to parse sentences with ellipsis, and in fact, probabilistic, neural, and Large Language Models (LLM) do so, too. We demonstrate experiments that focus on detecting sentences with ellipsis, predicting the position of elided elements, and predicting elided surface forms in the appropriate positions. We show that cross-linguistically reconstructing ellipsis and parsing it with SOTA NLP technologies results in acceptable representations for downstream tasks.

1 Introduction

As discussed in more detail from a typological perspective in (Cavar et al., 2024), ellipsis is a linguistic phenomenon that results in the omission of words in sentences that are usually obligatory in a given syntactic context and that the speaker and hearer can understand and reconstruct without effort.

While in discourse situations, different elements of utterances or sentences can be elided if they could be derived from the previous context, the constructions that we are interested in are ellipses in sentences without obligatory extra-sentential licensing conditions. A common ellipsis type that is

licensed within sentence boundaries is forward or backward conjunct reduction, as in example (1). It is common cross-linguistically. In the examples (1), the Croatian or German counterpart of *my sister* has been elided in the underlined position.

- (1) a. *Moja sestra* živi u Londonu i ___ radi u Amsterdamu. (Croatian)
- b. *Meine Schwester* lebt in London und ___ arbeitet in Amsterdam. (German)
- c. *My sister* lives in London and ___ works in Amsterdam.
- d. *My sister* lives in London and *my sister / she* works in Amsterdam.

The possibility of eliding phrases or words in coordinated constructions has universal and language-specific aspects. Certain ellipsis constructions are common in all languages we are aware of. Depending on underlying word order constraints, whether a language is an SVO or an SOV language results in language-specific ellipsis constraints. In addition, differences in morphology and general morphosyntactic properties can lead to peculiarities in the context of ellipsis.

Ellipsis constructions like FCR are possible in all languages we are aware of. In fact, whenever possible, ellipsis is the preferred form of presentation in text or spoken language in various construction types, e.g., in coordination constructions. This means that ellipsis is applied in unmarked cases whenever it is possible. We could hypothesize that ellipsis optimizes the signal entropy and improves communication by reducing time and effort. Whenever elements that could be elided remain overt in sentences or utterances, they might indicate specific semantic or pragmatic reasons. A sentence like (1d) appears to be emphatic if the phrase *my sister* is used. Such explicit repetitions of content stand in contrast to the unmarked default in ellipsis construction (1a).

In *gapping* constructions, as in (2a), we see that the verb complex *was watching* is elided. In example (2b), a case of VP-Ellipsis, the entire predicate or Verb Phrase (VP) *read War and Peace* is elided.

- (2) a. Paul and John were watching the news, and Mary ___ a movie.
 b. Susan read War and Peace but Mary did not ___

Ellipsis constructions like *gapping* do not require a licensing discourse context, i.e., no context outside of the sentence boundaries is necessary to license such ellipsis. Therefore, the licensing context is purely intra-sentential.

Discourse licensed ellipsis constructions are context-dependent and extra-sentential forms of ellipsis in responses to questions, as in example (3). The words *each candidate will talk* that are spelled out in the question (3a) are elided in the response (3b) (Cavar et al., 2024).

- (3) a. Will each candidate talk about taxes?
 b. No, ___ about foreign policy.

There are many more very specific ellipsis types that we cannot discuss in detail in this context. Each type of ellipsis comes with specific construction properties and limitations. One additional aspect of ellipsis worth mentioning here is that the elided content does not have to match the intra-sentential licensing context.

We can find some examples in English with lexical mismatches of elided word forms and licensing context, as in 4a. In the Croatian example (4b), a highly inflecting language, the licensing context morpho-syntactically and phonological does not match the elided forms. The elided content does not have to be homophonous with the intra-sentential licensing context. In the examples (4) the round brackets indicate the elided content and contain the morpho-syntactically correct forms that could fill the gaps.

- (4) a. John **reads** a book, but Paul and Mary (**read**) a newspaper.
 b. Ivan **je čitao** knjigu a Marija i Petar (**su čitali**) novine.
 I. be read book but M. and P. be read newspaper

A particularly problematic type of ellipsis is a scattered ellipsis of multiple words elided in different positions in a clause. In example (5) the

words *will*, *greet*, and *first* are elided in the second conjunct.

- (5) Will Jimmy greet Jill first, or ___ Jill ___ Jimmy ___ ?

As (Cavar et al., 2024) emphasized, and as pointed out in Testa et al. (2023) and Hardt (2023), common text genres exhibit a large number of all ellipsis types. Surprisingly, human processing is not at all impacted by ellipsis. On the contrary, ellipsis seems to improve the discourse and readability of text. The challenge for NLP processing such constructions is discussed below.

1.1 NLP Problems

One of the main issues why we experiment with ellipsis constructions is related to generating syntactic representations for subsequent semantic processing. In order to derive semantic representations and properties of utterances and sentences, we utilize functional relation annotation of sentence elements, for example, the automatic labeling of *subjects* and *objects*, or scope relations of quantifiers and operators. Common Dependency Grammar, Phrase Structure, or Lexical-functional Grammar parsers fail to analyze ellipsis constructions adequately. However, parsing ellipsis constructions with the elided elements undone or reconstructed results in significantly more useful parse trees. The examples (2a) and (2b) are not correctly parsed by common SOTA NLP-pipelines, while the examples (6a) and (6b) with ellipsis undone result in useful and acceptable parse trees.

- (6) a. Paul and John were watching the news, and Mary was watching a movie.
 b. Susan read War and Peace, but Mary did not read War and Peace.

Compare the parse trees generated by spaCy 3.7 using the English transformer model for (2a) and (6a) with the corresponding parse trees in Figures 1 and 2 respectively.

The problem with the DPT in Figure 1 is that the predicate head *watching* is coordinated with the direct object head in the second conjunct *movie*. At the same time, the subject in the second conjunct *Mary* is analyzed to be the subject of the direct object *movie*. With the ellipsis undone in the DPT in Figure 2, the dependency relations are correctly analyzed, resulting in a useful parse tree. These types of errors are systematic and can be replicated

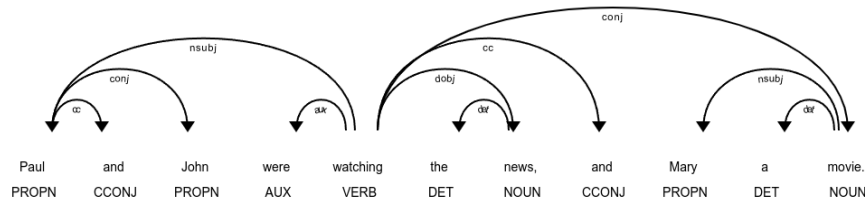


Figure 1: spaCy Dependency Tree (DPT) for example (2a).

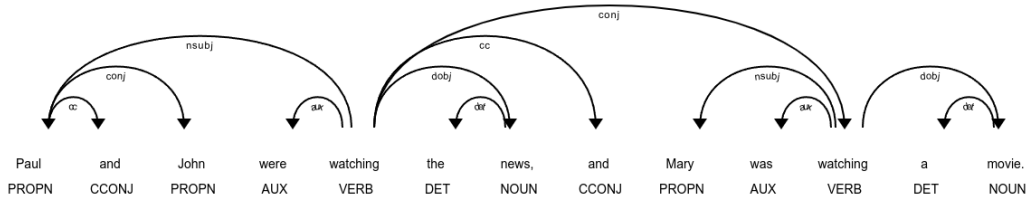


Figure 2: spaCy Dependency Tree (DPT) for example (6a).

for all our examples in The Hoosier Ellipsis Corpus (THEC) using various SOTA NLP pipelines.

Our experiments with parsing ellipsis constructions and comparing the output with ellipsis constructions undone were performed on the most recent versions of:

- Berkley Neural Parser (Benepar), (Kitaev and Klein, 2018; Kitaev et al., 2019), version 0.2.0, <https://github.com/nikitakit/self-attentive-parser>
- spaCy, (Honnibal and Johnson, 2015), version 3.7, <https://spacy.io>
- Stanza, (Qi et al., 2020), version 1.8.2, <https://stanfordnlp.github.io/stanza/>
- Xerox Linguistic Environment (XLE), (Crouch et al., 2011), <https://clarino.uib.no/iness/xle-web>

We experimented with all the languages in THEC for which we could identify models or grammar in the listed NLP pipelines. For almost all examples, the NLP pipelines generated inappropriate DPTs, Phrase Structure Trees (PST), or LFG style c- and f-structure pairs. Some of the error types are explained below.

In our evaluation of the NLP-pipeline output, the resulting trees were judged by a team of syntacticians familiar with all three relevant grammar frameworks, i.e., Dependency Grammar (DG), Phrase Structure Grammar (PSG), and Lexical-functional Grammar (LFG). Using spaCy, we were

able to experiment with Chinese, Croatian, English, German, Japanese, Korean, Norwegian, Polish, Russian, Spanish, and Swedish data. With XLE, we could only use the English, Norwegian, German, and Polish grammar. Stanza does not offer PST output for most of the languages we were targeting.

In the next section, we will describe how LLMs were as challenged with ellipsis constructions as these rule-based, statistical, or neural syntactic parsers.

While in most of the cases, the Dependency parser output improves with constructions without ellipsis, errors still remain problematic. Figure 3 shows an example with gapping of the verb in the second conjunct. The parser obviously confuses the coordination relation suggesting that the subject in the first conjunct *people* is coordinated with the auxiliary in the second conjunct *do*. However, the analysis of the first conjunct is already wrong since the predicate *like broccoli* is analyzed as a nominal modifier.

The error in the counterexample without ellipsis does not improve the parse tree in Figure 3. As the parse tree in Figure 4 shows, the conjunction relation is still wrong, suggesting that *people* and *like* is conjoined. The error in the first conjunct remains the same, while now the second conjunct structure without gapping results in an acceptable representation.

It is clear from a theoretical perspective that Dependency parsers will have issues with implicit lexical material in sentences. DG is primarily concerned with dependencies between overt lexical

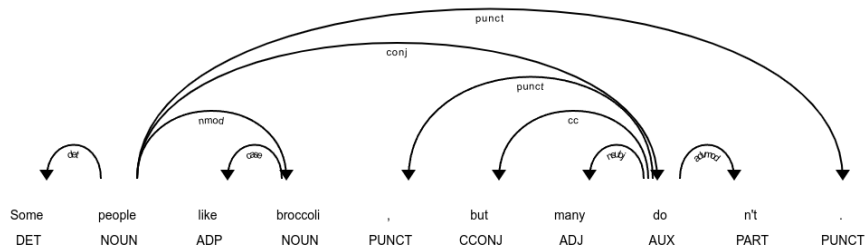


Figure 3: Stanza Dependency Tree Ellipsis 1

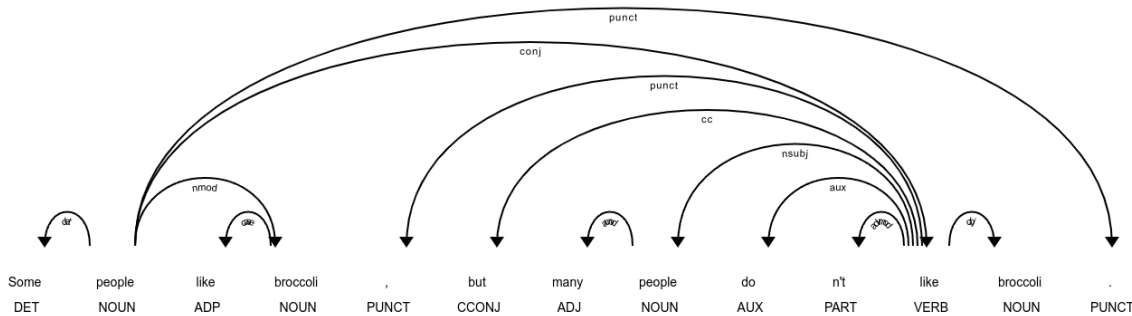


Figure 4: Stanza Dependency Tree Ellipsis 2

items and not implicit words or content.

The Stanza constituency parser does not provide a better result, as in Figure 5. It assumes the predicate head *like* to be the preposition head of a phrase modifying the subject phrase *some people*. The structure of the subordinate clause containing VP-ellipsis is useless for any further semantic post-processing.

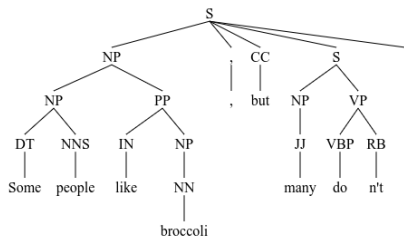


Figure 5: Stanza Constituency Tree

In our experiments, we can confirm that these examples are not rare parser errors. These are systematic mistakes that these parsers make in ellipsis constructions. The vast majority of ellipsis constructions will not be parsed correctly by current SOTA NLP pipelines, independent of the theoretical framework of the grammars or treebanks used for parser engineering, and independent of the parser model (e.g., rule-based, neural, LLM-based).

The following data and corpus creation and ex-

periments were motivated by the fact that document types like business reports, medical or technical documentation, as well as social media content, chat, or spoken language discourse, contain a large number of sentences with ellipses. Given that common SOTA NLP pipelines fail to provide adequate syntactic representations as tree structures, higher-level processing of discourse and semantic properties is not possible using their output.

As the example in (4) shows, morphologically rich languages allow lexically matching words to be elided, although the morpho-phonological surface form does not match. This does not seem to be a challenge for native speakers of these languages. However, it is a significant computational challenge to identify the correct morpho-phonological forms that were subject to ellipsis.

Scattered ellipsis, as in example (5), does not appear to be cognitively challenging, either; however, from a Machine Learning (ML) and NLP perspective, we expect to see significant errors and issues in identifying the ellipsis slots and guessing the elided words.

As mentioned above, intra-sentential licensing of ellipsis in gapping constructions is not necessarily dependent on the discourse context. Example 7 shows that complex gapping constructions are not restricted by syntactic phrase boundaries or structures, but maybe phonological conditions. None of

this is a parsing challenge for human listeners, but it is a significant problem for NLP pipelines.

- (7) Jimmy was always dreaming about going to Paris, and Mary ___ to Tokyo.

Our central goal in The Hoosier Ellipsis Corpus Project is to create corpora and language resources for the evaluation and development of NLP pipelines that can generate semantically more adequate syntactic structures for ellipsis constructions.

1.2 Previous Work

To present an overview of the theoretical work on ellipsis constructions in this context is impossible. Given the vast amount of publications on ellipsis using numerous descriptive and theoretical frameworks, we encourage the interested reader to consult excellent handbooks on that topic, for example, the Handbook of Ellipsis (van Craenenbroeck and Temmerman, 2018). The following summary focuses on recent computational and corpus approaches to ellipsis constructions.

Liu et al. (2016) investigated Verb Phrase Ellipsis (VPE) and conducted three tasks on two datasets. The first dataset is the Wall Street Journal (WSJ) section of the Penn Treebank with VPE annotation (Bos and Spenader, 2011), and the second dataset is sections of the British National Corpus annotated by Nielsen (2005) and converted by Liu et al. (2016) to the format used by Bos and Spenader (2011). The first task consisted of identifying the position of the element, called *target*, that is used to represent the elided verb phrase, called the *antecedent*. This first task only treats cases in which such a *target* is overtly present in the case of VPE, but this is not always the case, as shown in example 2b. The second and third tasks consisted of correctly linking the *target* to its *antecedent* and identifying the exact boundaries of the *antecedent*. Liu et al. (2016) found that the second and third tasks yielded better results when they were treated separately using two different learning paradigms rather than when they were treated jointly. They also found that a logistic regression classification model worked better for the first and third task, but that a ranking-based model yielded better results for the second task.

McShane and Babkin (2016) developed ViPER (VP Ellipsis Resolver), which is a system that uses linguistic principles, and more specifically syntactic features, to detect and resolve VP ellipsis. This

system is knowledge-based and does not use empirical data for training. It is not intended to solve all cases of VP ellipsis. It first detects the cases of VP ellipsis that are simple enough for the system to treat and then uses string-based resolution strategies. The system identifies the best *sponsor* string to fill and replace the elliptical gap. The system, evaluated against a GOLD standard dataset generated by the authors, had correctly resolved 61% of the VP ellipsis constructions it identified as simple enough to treat from the Gigaword corpus.

Droganova et al. (2018a,b) first created artificial treebanks containing elliptical constructions for English, Czech, and Finnish, using the Universal Dependencies (UD) (Nivre et al., 2016) annotation standard and evaluated several parsers in order to identify typical errors these parsers generate when dealing with elliptical constructions. Note that UD v2 used the *orphan* relation to attach the orphaned arguments to the position of the omitted element. The authors found that the F1-scores of most parsers were below 30%. This highlights how difficult it is for dependency parsers to identify elliptical constructions and warrants data enrichment for ellipsis resolution to improve dependency parsers' performances.

NoEI (An Annotated Corpus for Noun Ellipsis in English) was motivated by the assumption that noun ellipsis is more frequent in conversational settings. It is described in Khullar et al. (2020), where they annotated the first 100 movies of the Cornell Movie Dialogs dataset for noun ellipsis. Their annotation process involved using the Brat annotation tool to mark ellipsis remnants and their antecedents in the dataset. The dataset was manually annotated by three linguists, and an inter-annotator agreement was measured using Fleiss's Kappa coefficient, which indicated a high level of agreement among annotators. Their results show that a total of 946 cases of noun ellipsis existed in their corpus, corresponding to a rate of 14.08 per 10,000 tokens. The models they used included Naive Bayes, Linear and RBF SVMs, Nearest Neighbors, and Random Forest. They achieved an F1 score of 0.73 in detecting noun ellipsis using linear SVM and 0.74 in noun ellipsis resolution using Random Forest.

The Santa Cruz sluicing dataset is documented in Anand et al. (2021). In it, they compiled a corpus of 4,700 instances of sluicing in English, with each instance represented as a short text and annotated for syntactic, semantic, and pragmatic attributes. Most of the data they used comes from the New

York Times subcorpus of the English Gigaword corpus. The data set was created by identifying all verb phrases whose final child was a wh-phrase, and then manually culling false positives. Each of the instances is marked with five tags, namely, the antecedent, the wh-remnant, the omitted content, the primary predicate of the antecedent clause, and the correlate of the wh-remnant, if available.

The *ELLie* corpus and related experiments are discussed in [Testa et al. \(2023\)](#). It is a dataset of elliptical constructions that has been evaluated using GPT-2 ([Radford et al., 2019](#)) and BERT ([Devlin et al., 2019a](#)), two Transformer-based language models, on their ability to retrieve the omitted verb in elliptical constructions that demonstrate different levels of semantic compatibility between the missing element and its arguments. They found that while the performances of the two language models were influenced by the semantic compatibility of an elided element and its argument, these models had an overall limited mastery of elliptical constructions.

2 The Hoosier Ellipsis Corpus

The Hoosier Ellipsis Corpus (THEC) V 1.0 ([Cavar et al., 2024](#)) consists of data from eighteen languages. It includes data from low-resourced languages like Navajo and Kumaoni. To our knowledge, this is the only collection of ellipsis examples in some of these low-resourced languages. The THEC also contains unique collections of ellipsis constructions from common Slavic languages (Russian, Ukrainian, Polish).

The corpus includes the various ellipsis types, e.g., VP-ellipsis, Sluicing, Gapping, Stripping, Forward (FCR), and Backward Coordinate Reduction (BCR). Where necessary, the previous and following context of the ellipsis is provided as well.

While continuously extended with more data and other languages, [Table 1](#) lists the languages, and the current example counts in the THEC.

THEC data consists of sentence pairs. The Close test ([Taylor, 1953](#)) and the masked word machine learning approach taken in BERT ([Devlin et al., 2019b](#)) inspired the design of the data format. The example with ellipsis is provided, and the position of the elided content is marked with three underscores, as in [Figure 6](#). The fully spelled-out form of the corresponding ellipsis construction is separated by four dashes in a new line, providing the elided content.

Arabic	375	Croatian	6
English	267	German	79
Gujarati	9	Hindi	127
Japanese	105	Korean	40
Kumaoni	85	Mandarin Chinese	40
Navajo	9	Norwegian	55
Polish	139	Russian	202
Spanish	171	Swedish	20
Telugu	20	Ukrainian	158

Table 1: Corpus languages and example counts

Additionally, the example entry can be accompanied by the previous or following context. The previous context is indicated by the B: tag (for *before*), and the following context is indicated by the A: tag (for *after*). In a specific comment or meta information section of lines introduced by a hashmark, as in [Figure 6](#), the source of the example, the annotator, and a translation into different languages can be provided.

```
A Nina ___ na pianinie.
----
A Nina gra na pianinie.
B: Kasia gra na klarncie.
A: Marek śpiewa.
# source: Marjorie J. McShan (2000)
# TR eng: Nina plays piano.
```

Figure 6: Polish THEC gapping example with additional information.

This simple Unicode text-based format for encoding allows us to focus on common machine-learning approaches for experiments using various NLP technologies. This format also allows us to annotate ellipsis constructions that contain numerous elided slots (e.g., scattered ellipsis).

We focus in this data annotation approach on indicating the distributional properties of elided content in sentences, be it discourse licensed or purely syntactic ellipsis. The goal is to reflect the ‘understood’ or ‘implied’ sequence of words as understood by human native speakers, independent of any particular syntactic theory of ellipsis.

Our goal is to convert most of the ellipsis and full-form pairs into the Universal Dependencies 2 format with correctly encoded ellipsis.¹ In this simple data format, we can add PSG-style annotations

¹See for details the documentation for UD 2 at <https://universaldependencies.org/u/overview/specific-syntax.html>.

to the meta-section for every example, providing the phrase structure tree and additional syntactic information, or triple sets for dependencies, as well as c- and f-structure strings based on the LFG formalism.

The data source for the THEC is mainly literature and curated language corpora and data collections. We used mostly examples from peer-reviewed, theoretical, or documentary linguistic publications. In some cases we provide unique data that has not been published previously. In these cases the data was generated by native speakers (e.g., Navajo) and validated with their speaker communities.

3 NLP Experiments: Methods & Results

We reported in (Cavar et al., 2024) about the motivation for THEC and the first initial experiments testing NLP capabilities with the THEC constructions. Here, we expand these experiments to include new SOTA models and experimental strategies.

With the goal in mind to develop NLP pipelines that are capable of processing ellipses constructions and generating adequate representations, we defined three main tasks to test the capabilities of current SOTA NLP technologies and identify possible solutions for reconstructing fully spelled-out sentences from ellipsis constructions. The tasks involve a.) a binary classifier for the detection of ellipsis in sentences, b.) a model for the identification of the positions of elided content in sentences with ellipsis, and c.) a model for the prediction of the elided content in the correct positions in ellipsis constructions. The tasks a.) and b.) presuppose that the models are given only sentences that contain ellipses.

In (Cavar et al., 2024), we show that three different NLP approaches perform very differently and that LLMs were outperformed on task a.) by even a simple Logistic Regression classifier. The best-performing model for task a.) and task b.) was a BERT-based, Transformer-based classifier and labeler. For task c.), we could only utilize Large Language Models, of which only GPT-4 provided acceptable results for English, Spanish, and Arabic. In these initial experiments, we assumed that the Logistic Regression approach represents a baseline for the binary classification task but that it is less useful for guessing the positions of elided words and that it is useless in a task like c.), e.g., generating the morpho-syntactically correct word forms

for the elided content.

Initially, we expected transformer-based models to perform well as classifiers, we also expected them to be less efficient at guessing the position of elided content. Our expectation was also that current SOTA LLMs would be outperforming all other models in all three tasks. For generating the correct surface form of the elided content we did not see any other model beating SOTA LLMs since this is the natural task for Generative AI models.

3.1 Dataset

Using our manually compiled Ellipsis Corpus, we constructed three datasets. For English, we expanded the data with the ELLie corpus Testa et al. (2023), adding some corrections and modifications to it since some native speakers complained about the naturalness of some of the ellipsis constructions. We also used some sluicing examples from the Santa Cruz Sluicing dataset (Anand et al., 2018).

The first dataset was aimed at a simple binary classification task to detect and label sentences with 1 if they contain ellipsis and with 0 if not. The binary classification datasets were monolingual and a balanced mixture of target sentences and distractors. We generated a 10-fold randomized rotation of the examples to minimize any kind of sequencing effect when training classifiers or

Our corpus comprises pairs of examples showcasing ellipsis constructions, which specify both the location of the omitted element and the full form.

At this early stage of the Ellipsis Corpus, the languages that were represented with sufficient data were English, Russian, Ukrainian, Arabic, and Spanish. The experiments described in the following thus focus on these languages. We limit our description here to English and Arabic, since the format and results are almost completely equivalent to the settings for the other languages.

3.1.1 English Data

For English, we used 575 examples from ELLie and 559 examples from our manually compiled English Ellipsis Sub-Corpus. Combining each of the datasets with 658 distractor sentences, we generated a ten-fold randomized rotation of sentences.

For Task 1, the classification of ellipsis, we generated tuples with the sentence and label using the label 1 for ellipsis and 0 for no ellipsis.

For Task 2, we generated pairs of ellipsis and full-form sentences, leaving the underscore indi-

cators in the ellipsis example sentence to be able to train labeling algorithms that predict the ellipsis position or to evaluate predicted ellipsis positions directly.

3.2 Task 1: Binary Sentence Classification

The goal of Task 1 was to evaluate the performance of baseline approaches with transformer models and LLMs. As the baseline approach, we specified a simple Logistic Regression (LR) model that uses a sentence vectorization approach based on ten simple cues using linguistic intuition. For the generation of cue vectors for each sentence, we used the spaCy² NLP pipeline with the part-of-speech tagger and Dependency parser. The classification vectors for each English sentence were generated using the following information: the number of nouns; the number of subject dependency labels; the number of object dependency labels; the number of conjunctions; the number of *do so*; a boolean whether a *wh*-word is sentence-final; the number of verbs; the number of auxiliaries; the number of *acom* Dependency labels; the number of tokens *too*.

We trained a binary LR classifier using these ten-dimensional vectors. The goal was not to optimize the classifier and achieve the best possible result but to develop a simple baseline classifier using just a few linguistic cues for ellipsis constructions.

The transformer-based classifier is based on BERT for English.

For GPT-4 we used context *Classify the following sentence as containing ellipsis or not. Ellipses indicates gapping, pseudogapping, stripping, and sluicing. Answer with only 0 for sentences without ellipsis or only 1 for sentences with ellipsis.* which preceded each sentence.

We additionally conduct few-shot experiments on GPT-4 in which the model is given 4 example annotations in addition to its prompt. These examples are omitted when calculating results.

3.3 Task 2: Locate of Ellipsis

In this task, we evaluate Language Models and specific transformer models with respect to their ability to predict the precise location of elided words. The complexity in this task varies from one elided word, multiple elided words as in example (7), and scattered multi-slot ellipsis as in example (5).

The data set for this task consists of sentence

²See <https://spacy.io/> for more details.

pairs. One sentence contains the indicators (3 underscores) for the ellipsis positions, while the other one does not contain such indications and is used for testing the models. The models are trained and tested only using examples that contain ellipses. Ten-fold random rotations of examples are tested on BERT-based sequence labeling.

For GPT-4 we used a prompt with a rich context: *Annotate the following sentence by placing ___ in the position of each ellipsis. Ellipses indicates gapping, pseudogapping, stripping, and sluicing. If there are no ellipses, answer with only original sentence.* We additionally conduct few-shot experiments on GPT-4. Accuracy is calculated by comparing the correctly annotated sentence to the generated GPT-4 sentence.

3.4 Task 3: Generate Elided Words

In this task, we evaluate LLMs for their ability to generate the elided word in the correct positions. The data set consists of sentence pairs. One of the sentences contains ellipsis and the other is the "full-form" of the same sentence with the elided words spelled out. Only examples with ellipses were used for training and testing the models.

For the GPT-4-based evaluation, we used a prompt with a rich context: *Insert any missing words implied by ellipses. Ellipses indicates gapping, pseudogapping, stripping, and sluicing. Answer with only the new sentence. If there are no ellipses, answer with only the original sentence.* We additionally conduct few-shot experiments on GPT-4.

4 Results

We tested GPT-4 (gpt-4-turbo-2024-04-09) zero-shot and few-shot, BERT, and LR. Alongside our binary LR classifier, we tested GPT-4 (gpt-4-turbo-2024-04-09) and BERT. For GPT-4, we tested on our dataset of English, Arabic, and Spanish. The results for task 1 are given in Table 2.

Model/Language	en	es	ar
LR	0.74	-	-
BERT	0.94	-	-
GPT-4 zero-shot	0.59	0.73	0.61
GPT-4 few-shot	0.64	0.75	0.73

Table 2: Task 1 Binary Classification Accuracy for English, Spanish, and Arabic

It is surprising that the GPT-4 zero-shot classi-

fication is worse than the LR-baseline, and significantly worse than the BERT-based classifier. Precise scores from the zero-shot and few-shot GPT-4 experiments are given in Table 3.

GPT-4 zero-shot			
Language	f1	p	r
English	0.63	0.54	0.76
Spanish	0.70	0.60	0.85
Arabic	0.33	0.33	0.33
GPT-4 zero-shot			
Language	f1	p	r
English	0.65	0.60	0.70
Spanish	0.70	0.63	0.79
Arabic	0.67	0.52	0.94

Table 3: Precision, Recall, and F1-Score for GPT-4 across English, Spanish, and Arabic

Given the default temperature setting of 0.7 in GPT-4, the output from the model is not deterministic for a given input sentence. In order to reduce randomness in the model, we set the temperature of GPT-4 to 0. This approximates the model choosing a response that it deems most probable, instead of it sampling from possible responses.

In Task 2, we tested an initial BERT-based ellipsis position guesser and GPT-4 zero-shot and few-shot. Task 2 results are shown in Table 4.

model/language	en	es	ar
BERT	0.70	-	-
GPT-4 zero-shot	0.18	0.27	0.07
GPT-4 few-shot	0.26	0.34	0.15

Table 4: Task 2 Ellipses Location Identification Accuracy for English, Spanish, and Arabic

Surprisingly, BERT achieved an accuracy of 0.7. The GPT-4-based experiments on Task 2 were challenging. The prompt engineering for the zero-shot experiment resulted in a low accuracy across all languages. For Task 3, we exclusively focused on the evaluation of GPT-4. Results for Task 3 are shown in Table 5.

model/language	en	es	ar
GPT-4 zero-shot	0.22	0.29	0.01
GPT-4 few-shot	0.34	0.42	0.35

Table 5: Task 3 Elided Word Generation Accuracy for English, Spanish, and Arabic

GPT-4 performed better on elided word genera-

tion than ellipses location identification, however this remained a difficult task with low accuracy across all languages. In all tasks, few-shot improved GPT-4 performance.

5 Conclusion

Ellipsis constructions are obviously still challenging for all the common SOTA NLP pipelines and rule-based systems. Use of Dependency or Constituency parse trees, or even LFG c- and f-structures for syntactic and semantic processing of real-world data from different genres or registers is limited due to the fact that ellipsis is a common and widespread phenomenon in all languages.

The problem can be partially linked to grammar frameworks like Dependency Grammar or LFG, which do not necessarily foresee opaque linguistic elements (e.g., elided words or phrases) to be active rule elements modeled in grammar rules or descriptive formal annotation frameworks. While UD provides the instruments for annotating or handling ellipses, those instruments need to be more extensive for the description of the different intra- and cross-linguistic ellipses types. We also suspect that parsing algorithms and the training of parsers need to include such opaque elements and potentially new learning strategies.

The fact that specific models trained on the prediction of ellipses in sentences outperform LLMs seems to indicate that the lack of explicit data and pure self-supervised machine learning is not sufficient to handle opaque elements in language, either. Training LLMs on purely overt data ignores significant properties of language. Ellipsis phenomena are grammatical and systematic, and it seems problematic for current LLMs to guess covert continuations.

Given that there is too little data on ellipsis in general and none at all for most languages, it seems necessary to continue our Ellipsis Corpus project and provide not only sufficient data for the different languages but also a good typological overview of the different manifestations of ellipsis phenomena in different languages and language groups.

The Ellipsis Corpus and the relevant code for the experiments described in the article are available on GitHub: <https://github.com/dcavar/hoosierellipsis Corpus>.

References

- Pranav Anand, Daniel Hardt, and James McCloskey. 2021. The santa cruz sluicing data set. *Language*, 97(1):e68–e88.
- Pranav Anand, Jim McCloskey, and Dan Hardt. 2018. Santa cruz ellipsis consortium sluicing dataset (1.0).
- Johan Bos and Jennifer Spenser. 2011. An annotated corpus for the analysis of vp ellipsis. *Language resources and evaluation*, 45:463–494.
- Damir Cavar, Ludovic Mompelat, and Muhammad Abdo. 2024. [The typology of ellipsis: A corpus for linguistic analysis and machine learning applications](#). In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 46–54, St. Julian’s, Malta. Association for Computational Linguistics.
- Richard Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell, III, and Paula Newman. 2011. *XLE Documentation*. Xerox Palo Alto Research Center, Palo Alto, CA.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). pages 4171–4186.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kira Drohanova, Filip Ginter, Jenna Kanerva, and Daniel Zeman. 2018a. Mind the gap: Data enrichment in dependency parsing of elliptical constructions. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 47–54.
- Kira Drohanova, Daniel Zeman, Jenna Kanerva, and Filip Ginter. 2018b. Parse me if you can: Artificial treebanks for parsing experiments on elliptical constructions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Daniel Hardt. 2023. [Ellipsis-dependent reasoning: a new challenge for large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 39–47, Toronto, Canada. Association for Computational Linguistics.
- Matthew Honnibal and Mark Johnson. 2015. [An improved non-monotonic transition system for dependency parsing](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.
- Payal Khullar, Kushal Majmundar, and Manish Shrivastava. 2020. Noel: An annotated corpus for noun ellipsis in english. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 34–43.
- Nikita Kitaev, Steven Cao, and Dan Klein. 2019. [Multilingual constituency parsing with self-attention and pre-training](#).
- Nikita Kitaev and Dan Klein. 2018. [Constituency parsing with a self-attentive encoder](#).
- Wolfgang Klein. 1981. Some rules of regular ellipsis in german. In W. Klein and W.J.M. Levelt, editors, *Crossing the Boundaries in Linguistics. Studies Presented to Manfred Bierwisch*, pages 51–78. Reidel, Dordrecht.
- Zhengzhong Liu, Edgar González, and Dan Gillick. 2016. Exploring the steps of verb phrase ellipsis. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016), co-located with NAACL 2016*, pages 32–40, San Diego, California. Association for Computational Linguistics.
- Marjorie McShane and Petr Babkin. 2016. [Detection and resolution of verb phrase ellipsis](#). *Linguistic Issues in Language Technology*, 13.
- Leif Arda Nielsen. 2005. *A corpus-based study of verb phrase ellipsis identification and resolution*. Ph.D. thesis, Citeseer.
- Joakim Nivre et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A Python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Wilson L. Taylor. 1953. [“cloze procedure”: A new tool for measuring readability](#). *Journalism Quarterly*, 30(4):415–433.
- Davide Testa, Emmanuele Chersoni, and Alessandro Lenci. 2023. We understand elliptical sentences, and language models should too: A new dataset for studying ellipsis and its interaction with thematic fit. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers*, pages 3340–3353. Association for Computational Linguistics.
- Jeroen van Craenenbroeck and Tanja Temmerman. 2018. *The Oxford Handbook of Ellipsis*. Oxford University Press.

Modeling morphosyntactic agreement as neural search: a case study of Hindi-Urdu

Alan Zhou
Johns Hopkins University
azhou23@jhu.edu

Colin Wilson
Johns Hopkins University
colin.wilson@jhu.edu

Abstract

Agreement is central to the morphosyntax of many natural languages. Within contemporary linguistic theory, agreement relations have often been analyzed as the result of a structure-sensitive search operation. Neural language models, which lack an explicit bias for this type of operation, have shown mixed success at capturing morphosyntactic agreement phenomena. This paper develops an alternative neural model that formalizes the search operation in a fully differentiable way using gradient neural attention, and evaluates the model’s ability to learn the complex agreement system of Hindi-Urdu from a large-scale dependency treebank and smaller synthetic datasets. We find that this model outperforms standard architectures at generalizing agreement patterns to held-out examples and structures.

1 Introduction

Agreement is central to the morphosyntax of many natural languages (e.g., [Moravcsik, 1978](#); [Corbett, 2006](#); [Baker, 2008](#)). For example, in Hindi-Urdu sentences such as (1), the main verb and auxiliary agree in number and gender with the subject (as indicated by **bold**; examples here from [Bhatt, 2005](#)).¹

- (1) **Rahul** kitaab **paRh-taa** **thaa**
Rahul.M book.F read-Hab.MSg be.Pst.MSg
Rahul used to read (a/the) book.

Across languages, agreement systems are sensitive to a wide yet restricted range of properties: grammatical categories and features such as Case, grammatical functions such as subject and object, structural positions such as specifier and complement, syntactic relations of dominance and c-command, as well as syntactic locality (shortest-path node distance). Agreement is also distinguished by being ‘fallible’ ([Preminger, to appear](#)): when no suitable

controller for agreement exists, the target can take on default features (e.g., masculine singular).

Verb agreement in Hindi-Urdu illustrates much of this complexity. For example, in (2), the verb and auxiliary agree with the Nominative object instead of the Ergative subject (cf. the Nominative subject in (1)). In (3), verb agreement ‘fails’ because the subject and object both have overt Case (Ergative and Accusative). Most strikingly, Hindi-Urdu allows ‘long-distance’ agreement (LDA) as in (4): when all of the local noun phrase arguments have overt Case marking, a verb can agree with the Nominative object of an embedded clause.

- (2) Rahul ne **kitaab** **paRh-ii**
Rahul.M Erg book.FSg read-Pfv.FSg
thii
be.Pst.FSg
Rahul had read (a/the) book.
- (3) Rahul ne kitaab ko paRh-aa
Rahul.M Erg book.F Acc read-Pfv
thaa
be.Pst.MSg
Rahul had read the book.
- (4) Vivek ne [**kitaab parh-nii**]
Vivek.M Erg book.F read-Inf.F
chaah-ii
want-Pfv.FSg
Vivek wanted to read the book.

In this paper, we develop a neural model of morphosyntactic agreement that is capable of representing intricate agreement systems like those attested cross-linguistically, and evaluate its ability to learn the system of Hindi-Urdu from a large dependency treebank as well as much smaller synthetic datasets. We begin by situating our model in the context of morphosyntactic theory and previous computational approaches to agreement. Following many contemporary theoretical proposals, our model formalizes agreement as structure-dependent *search* from targets (*probes*) to controllers (*goals*). As in

¹Example sentences provided throughout the paper follow the glossing and transliteration of the original sources.

some previous models, agreement is implemented with soft neural *attention* and other differentiable mechanisms, rather than by symbolic tree traversal and feature copying.

2 Related research

2.1 Morphosyntactic theory

In some contemporary linguistic theories, agreement is a fundamental structure-building operation of syntax (e.g., Chomsky, 1995; Deal, 2015). In others, agreement is treated as postsyntactic: a part of morphology that operates on fully-formed syntactic structures (e.g., Bobaljik, 2008). Within both approaches, there is broad consensus that agreement relations are established by tree-based *search* (e.g., Preminger, to appear; Baker, 2008; Ke, 2023).

The details of the search operation remain controversial. Preminger (to appear) argues for strictly serial and ‘downward’ search in which each agreement probe explores the nodes of its c-command domain in a preset order and halts when it finds a suitable goal — or fails to find a goal before reaching terminal and blocking ‘phase’ nodes (resulting in default agreement). Others argue for different directionality, allowing a probe to optionally or obligatorily look ‘upwards’ to nodes that c-command it (e.g., Bjorkman and Zeijlstra, 2019; Baker, 2008). Still others argue for more elaborate operations that can occur as part of the search (Béjar and Rezac, 2009; Deal, 2015), or propose alternative conditions under which search halts (Deal, 2015).

The neural model that we propose is postsyntactic, insofar as it takes complete syntactic structures as inputs, but is otherwise compatible with many theoretical frameworks and varieties of search. We assume minimally that input structures consist of nodes, that nodes are specified for grammatical category (e.g., noun vs. verb), that some nodes have specifications for phi-features (e.g., person, number, gender) and other morphosyntactically relevant properties such as Case (e.g., Nominative vs. Ergative), that some nodes are designated as agreement probes (or as having ‘uninterpretable’ phi-features to be satisfied by agreement), and that nodes enter into (labeled) syntactic relations of dominance or dependency with one another. The model is architecturally agnostic about search directionality and our application to Hindi-Urdu uses both ‘downward’ and ‘upward’ probing.

2.2 Neural models

Previous computational research has explored whether recurrent neural networks (RNNs) and transformer models can capture morphosyntactic agreement (Linzen et al., 2016; Li et al., 2023; Bacon and Regier, 2019; Goldberg, 2019), with mixed success. Evaluating on English subject-verb agreement, Linzen et al. (2016) find that RNNs require explicit supervision of verb inflection to approximate structure-sensitive dependencies, despite seemingly high accuracy when trained only on a language modeling task. More robust sensitivity to structure is found for transformer architectures (Goldberg, 2019; Wilson et al., 2023), though these models are still not entirely unaffected by non-goal ‘distractors’ and are more susceptible to linearly close distractors than humans.

Previous models further struggle to capture agreement dependencies for languages with more complex agreement phenomena. Ravfogel et al. (2018) find that recurrent neural networks have difficulty learning the agreement system of Basque, in which auxiliary verbs agree with several local arguments, instead showing some reliance on surface heuristics instead of syntactic structure. A cross-linguistic evaluation of transformers (Bacon and Regier, 2019), following (Goldberg, 2019), finds that transformers struggle significantly with agreement in a handful of languages, such as Persian, Basque, and Finnish, as well as noting their sensitivity to distractors even when performance is overall high.

Similar results have been found for verb agreement in French (Li et al., 2023). Evaluating an RNN and a transformer on two different agreement patterns in French, the authors find that both models achieve relatively high accuracy. However, they see a degradation in performance when surface heuristics — such as agreement with the linearly first or most recent noun phrase — fail to predict the correct inflection. Additionally, while the attention patterns of the transformer model indicate that it appropriately distinguishes the two agreement patterns, the sensitivity to heuristics makes attention difficult to interpret in a syntactically coherent way.

A separate line of work explores models that explicitly learn agreement rules. Chaudhary et al. (2020) use a decision tree to extract rules predicting agreement across multiple languages in the Universal Dependencies family of treebanks (Nivre

et al., 2020). While this works well for certain languages like Greek or Russian, performance varies widely from language to language and especially drops in ‘zero-shot’ settings with minimal training data. Importantly, this model operates only between nodes that are directly connected within a dependency tree, making it unable to capture long-distance agreement as in example (4) above.

Our contribution shares high-level aspects of these proposals, including the use of continuous embeddings and attention, but differs in its goals and scope. We do not treat morphosyntactic agreement as a language modeling problem, recurrent or otherwise, but rather follow syntactic theory in taking agreement to be essentially a (postsyntactic) relation among syntactic nodes.

The model that we propose establishes these relations through search — technically, iterative redistribution of attention among nodes — conditioned on the types of morphosyntactic relations and features that are relevant for agreement cross-linguistically. The model does not parse sentences or generate inflected wordforms: it is designed solely to capture agreement but, in virtue of being fully differentiable, could be incorporated into larger neural models for parsing, inflection, or other applications. It has a small number of trainable parameters that can be set for particular agreement patterns, such as that of Hindi-Urdu.

3 Agreement in Hindi-Urdu

Agreement in the language of our case study has been extensively investigated within descriptive and theoretical linguistics (e.g., Pandharipande and Kachru, 1977; Bhatt and Keine, 2017; Mohanan, 1994; Bhatt, 2005; Kachru, 1970; Butt, 1993). A generalization that covers all of the examples in (1) - (4) is that Hindi-Urdu verbs and auxiliaries in the matrix clause agree in gender and number with *the highest non-overtly Case-marked noun phrase*, where all Cases other than Nominative/Absolutive are overt.

The notion of ‘highest’ can be defined in many technical ways (e.g., in terms of proximity to a Tense or Inflection node), but basically tracks the well-known accessibility hierarchy subject > direct object > indirect object > other (e.g., Moravcsik, 1978; cf. Bobaljik, 2008). When there is no such noun phrase, masculine singular is used by default.

Hindi-Urdu is particularly remarkable for allowing long-distance agreement (LDA), and for the

intricacies of agreement in light-verb constructions. Below we provide some further details about each of these phenomena, both of which occur in the datasets used to evaluate our model. For a more comprehensive view of Hindi-Urdu agreement and morphosyntax, we refer readers to original sources (e.g., Bhatt, 2005; Butt, 1995; Mohanan, 1994).

3.1 Long Distance Agreement

As illustrated in (4), verbs and auxiliaries can agree with non-overtly Case marked arguments of infinitival embedded clauses when no ‘higher’ noun phrase is suitable. This agreement is optional: (5) below, which differs from (4) in that both the matrix and embedded verbs show default agreement, is also acceptable. Mahajan (1990) notes some interpretation differences between these cases, in which LDA seems to make the object more ‘specific’ (examples below based on Bhatt, 2005).

- (5) Vivek ne [kitaab parh-naa]
 Vivek.M Erg book.M read-Inf.M
 chaah-aa
 want-Pfv.MSg
 Vivek wanted to read the book.

Bhatt (2005) also notes a parasitism in LDA, such that the matrix and embedded infinitival verb must either both agree with the same noun phrase or both take default features. Neither (6a), which has infinitival agreement without LDA, nor (6b), which has LDA but not infinitival agreement, is acceptable according to that source.

- (6) a. *Shahrukh ne [tehnii kaat-nii]
 Shahrukh Erg branch.F
 chaah-aa
 cut-Inf.F want-Pfv.MSg
 Shahrukh had wanted to cut the branch.
 b. *Shahrukh ne [tehnii kaat-naa]
 Shahrukh Erg branch.F cut-Inf.M
chaah-ii thii
 want-Pfv.F be.Psts.FSg
 Shahrukh had wanted to cut the branch.

However, this parasitism may be dialect specific. Butt (1993) provides the following example in which the infinitival verb agrees with its embedded object but the matrix verb agrees with its Nominative subject.

- (7) Ram [rotii khaa-nii] caah-taa
 Ram.M bread.F eat.Inf.FSg want-Impf.M.Sg
thaa
 was
 Ram wanted to eat the bread.

Parasiticism motivates Bhatt to propose an additional operation that allows a probe to create dependencies between heads as part of the search process. We do not formalize this extra mechanism here, and therefore focus on Butt’s dialect, which is consistent with the root and infinitival verbs being separate probes. Parasitic agreement should be addressed by future elaborations of the model.

3.2 Light Verb Agreement

Light-verb constructions make up a majority of verbal predications in the language (e.g., Ahmed et al., 2012; Vaidya et al., 2019, 2016). In these constructions, a semantically less meaningful *light* verb (e.g. *kar* ‘do’, *ho* ‘be’) combines with a more meaningful noun, verb, or adjective (example from Ahmed et al., 2012).

- (8) a. **NAdiyah** hans **paR-I**
 Nadiya.F.Sg laugh fall.Perf.F.Sg
 Nadya burst out laughing.
- b. YAsIn nE **mEz** s3Af
 Yasin.M.Sg Erg table.F.Sg clean
k-I
 do.Perf.F.Sg
 Yasin made the table clean.

Agreement morphology in these constructions is always on the light verb. In both the V-V (8a) and Adj-V (8b) constructions, agreement follows from the same generalizations discussed earlier. However, a somewhat different pattern is found in N-V light verb constructions (examples from Mohanan, 1994):

- (9) a. Ilaa ne mohan kii **prasamsaa**
 Ila Erg Mohan Gen praise.F
kii.
 do.Perf.F
 Ila praised Mohan.
- b. Ilaa ne **kissaa** yaad
 Ila.F Erg incident.M memory.F
kiyaa.
 do.Perf.M
 Ila remembered the incident.
- c. Ilaa ne Mohan ko yaad
 Ila Erg Mohan Acc memory.F
 kiyaa
 do.Perf.M
 Ila remembered Mohan.

Unlike for Adj-V and V-V, members of one class of nouns in N-V constructions are eligible for agreement, as shown in (9a). When conjoined with a light verb, these nouns select either an object with

oblique Case (e.g., Genitive in (9a)), or no object at all (Mohanan, 1994). Members of another class of nouns do not agree in N-V constructions, as in (9b, 9c). These form a predicate that selects for a direct Case (Nominative, Accusative, or Ergative) object, and agreement patterns follow as expected.

LDA and light-verb constructions can occur together. For example, in (10) the embedded infinite clause contains an N-V predicate. Both the matrix and embedded verbs agree with the noun component of the light verb (example from Bhatt, 2005).

- (10) Akbar ne [meri **madad** **kar-nii**] **chaahii**
 Akbar Erg my.F help.F do.Inf.F want.Pfv.F
thii
 be.pst.FSg
 Akbar had wanted to help me.

4 Model

The neural model that we propose takes as input a syntactic tree, with certain nodes designated as agreement probes, and outputs predicted phi-feature values for each probe. Here we apply the model to Hindi-Urdu dependency trees (Bhat et al., 2017; Palmer et al., 2009) and synthetic trees based on those (see section 5.2.2). The edges between nodes are therefore directed and labeled by UD relations Nivre et al. (2020, e.g., nsubj, obj, aux). Future research could experiment with constituency trees of the type that are more familiar in generative syntax, perhaps with minimal labeling of edges (e.g., specifier vs. complement).

Below we describe our neural embedding of dependency trees, the search process that distributes attention from probes to goals (or defaults), the transfer of predicted features to probes, as well as the loss function and other model details. We also describe two baseline transformer models, and compare the performance of our model to those on learning Hindi-Urdu verb agreement.

4.1 Tree embedding

The N nodes of a given syntactic tree are assumed to be arbitrarily ordered (n_0, n_1, \dots) and represented as feature vectors with the minimal cross-linguistically motivated content. Specifically, separate one-hot vectors are used to embed grammatical category (e.g., noun, verb, auxiliary), each phi-feature separately (e.g., person, gender, number), and Case (e.g., Nominative, Accusative, Ergative). Zero vectors are used for unspecified features (e.g., root verbs are not specified for Case). These

vectors are stacked into a single embedding \mathbf{f}_i for each node n_i , and the embeddings are arranged as rows in a matrix \mathbf{F} following the arbitrary node order. Each node also has a separate one-hot embedding \mathbf{d}_i of the dependency relation that it bears with its (unique) parent, and these are likewise arranged as rows in a matrix \mathbf{D} .

To facilitate our search algorithm, two minor modifications are introduced for each tree. First, we create a ‘self’ connection from each node to itself that bears its own special dependency relation. This gives the model the option to ‘stay’ at a node during the search process, rather than being forced to pick from one of its neighbors. Additionally, we introduce a ‘default’ node to each tree that has in-going connections from every other node, but an out-going connection only to itself. This node is entirely featureless in terms of phi-features, part-of-speech, and Case during the search process, but is associated with default phi-features during the feature valuation step of the model (see below).

Because dependency relations are embedded as properties of child nodes, including edge labels would be redundant. Therefore, the edges of a tree are represented with a binary adjacency matrix \mathbf{H} , where $H_{ij} = 1$ indicates that node n_i is the head of node n_j . The transposed adjacency matrix \mathbf{H}^T relates dependents in rows to heads in columns.

4.2 Searching from probes to goals

Each designated probe in a tree searches for a goal with which to agree by initially attending to itself and then iteratively redistributing attention to other nodes in the tree. The single-step redistribution of attention is determined by a stochastic transition matrix conditioned on the topology of the tree and learnable weight vectors via the softmax function. Multiple-step search simply iterates the same transition matrix for a fixed topology and weights.

Within a language, probes seek goals that bear particular features and dependency relations. We formalize this with two weight vectors \mathbf{w} (of the same dimensionality as each \mathbf{f}_i) and \mathbf{v} (of the same dimensionality as \mathbf{d}_i). The latter weights the ‘downward’ direction of dependencies — from heads to their dependents. To independently weight the ‘upward’ direction — from dependents to their heads — we use another vector \mathbf{u} . The model has two additional scalar weights, w_{self} and $w_{default}$, which correspond to self and default node dependencies as described above.

Each node assigns a logit score to its dependents

on the basis of their features and their relations. These scores are represented in the $N \times N$ matrix \mathbf{S}_{down} as defined below. Similarly, each node assigns a logit score to its parent and these are collected in the $N \times N$ matrix \mathbf{S}_{up} . Finally, each node also assigns a score to itself according to the self dependency, represented in \mathbf{S}_{self} . In our notation, \odot is the elementwise (Hadamard) product and common broadcasting conventions are assumed.

$$\begin{aligned} \mathbf{S}_{down} &= \mathbf{H} \odot [\underbrace{(\mathbf{F} \mathbf{w})^T}_{1 \times N} + \underbrace{(\mathbf{D} \mathbf{v})^T}_{1 \times N}] \\ \mathbf{S}_{up} &= \mathbf{H}^T \odot [\underbrace{(\mathbf{F} \mathbf{w})^T}_{1 \times N} + \underbrace{(\mathbf{D} \mathbf{u})}_{N \times 1}] \\ \mathbf{S}_{self} &= \mathbf{I}_N \odot [\underbrace{(\mathbf{F} \mathbf{w})^T}_{1 \times N} + w_{self}] \\ \mathbf{S} &= \mathbf{S}_{down} + \mathbf{S}_{up} + \mathbf{S}_{self} \\ \hat{A}_{ij} &= \begin{cases} S_{ij} & \text{if } S_{ij} \neq 0 \\ -\infty & \text{if } S_{ij} = 0 \end{cases} \\ \mathbf{A}_i &= \text{softmax}(\hat{\mathbf{A}}_i) \end{aligned}$$

The i th row of the $N \times N$ matrix \mathbf{S} contains the logit scores that node n_i assigns to every other node n_j with which it is related by dependency (including self-dependency and the default node). To convert these into probabilities, we mask out zero entries of \mathbf{S} and take the row-wise softmax to derived the single-step transition matrix \mathbf{A} .

Note that the zero-one encoding of adjacencies in \mathbf{H} and \mathbf{I}_N ensure that the transition probabilities of \mathbf{A} are only non-zero from nodes to their immediate neighbors (including the default node). Additionally, the default node has a transition probability of 1 to itself (hence 0 to all other nodes).

Let \mathbf{p} be an N -dimensional binary vector that indicates which nodes of the tree are probes (with a final zero element for the default). The search process begins with each probe node attending fully to itself with a one-hot vector at its own position, as stated in the definition of $\mathbf{P}^{(0)}$. Search then proceeds — attention in each row is iteratively re-allocated — simply by multiplying the previous \mathbf{P}^{t-1} with \mathbf{A} .

$$\begin{aligned} \mathbf{P}^{(0)} &= \mathbf{I}_{N+1} \odot \mathbf{p} \\ \mathbf{P}^t &= \mathbf{P}^{t-1} \mathbf{A} \end{aligned}$$

Observe that \mathbf{A} is constant for a given tree and weights, and can therefore be precomputed prior

to search by all probes in the tree. Observe further that rows of \mathbf{P}^t for non-probe nodes are identically zero; these could be ignored in sparse matrix implementations.

The search process is repeated for a fixed number of steps t_{max} , allowing a probe to iteratively explore the tree from its starting position. At the end of search, we take the final attention scores of a probe to be a distribution over the goal nodes that a probe ‘returns.’ The entire search can thus be viewed as a Markov process, with the nodes of a tree being the states over which the transition matrix operates (e.g., the default node is an absorbing state).

Intuitively, our formalization results in a *gradient breadth-first search*. Note that our structurally-informed transition matrix ensures that for any individual step, attention can only be reallocated from a node to itself or its immediate neighbors. Thus, at step t , each probe’s attention can only be allocated among nodes that are at most t steps away from its probe node. We additionally observe that after learning this process converges to an approximation of *greedy search*, in which attention for a given probe is nearly one-hot at each step.

4.3 Feature Valuation

The features that are copied to the probe are the weighted sum of phi-features from each node the probe attends to. To compute this, we construct a phi-feature matrix \mathbf{E}_ϕ , whose i th row contains the concatenation of n_i ’s one-hot phi-feature embeddings, or the concatenated phi-feature embeddings for a language’s default phi-features (masculine singular for Hindi-Urdu) if n_i is the default node. This results in a $N \times D_\phi$ matrix, where D_ϕ is the dimensionality of our concatenated embeddings.

The predicted features for a probe are then the result of multiplying $\mathbf{P}^{(t_{max})}$ by \mathbf{E}_ϕ :

$$\mathbf{Y}_{pred} = \mathbf{P}^{(t_{max})}\mathbf{E}_\phi$$

4.4 Objective

During training, the model’s predicted features are compared with the correct phi-features on each probe node by cross-entropy loss. Assuming perfect annotation of phi-features on probes and goals, this can be done directly. However, in our naturalistic treebank, many lexical items that are not overtly inflected for phi-features are mislabeled as having null phi-features (e.g. proper nouns and certain auxiliaries that do not inflect for gender). To

account for this, we take the argmax of the one-hot feature predictions as the discrete ‘prediction’ for a probe, and mask out the parts of the cross-entropy loss where either this prediction or the true feature value is null. We similarly use the argmax at test time to determine the predicted phi-features that each probe returns.

5 Evaluation

We trained our model on both naturalistic data from the Hindi UD treebank and synthetic data from a hand-designed dependency grammar. As noted above, we assume the dialect from Butt (1993), which does not require a probe to additionally create dependencies during its search. Therefore, we initialized a probe at each verb and auxiliary. A modest value of $t_{max} = 3$ steps was found to be sufficient for these data sets. To test our model’s structural generalization ability, we also increased this to $t_{max} = 5$ on a relative clause distractor task (see section 5.2.2 below).

5.1 Transformer Baselines

We compared our model, referred to below as Search, against two transformer baselines: a Cloze transformer that predicts the phi-features of masked-out probes given the entire sentence, and a language model (LM) transformer that predicts the phi-features of masked-out probes given the preceding tokens in a sentence. These two transformer models are identical in architecture, featuring a one-head, one-layer transformer encoder, followed by a linear decoder that maps each token’s embedding to a phi-feature prediction.²

Linearized (surface order) trees were used as inputs to these models, with each token embedded by stacking one-hot vectors for its part-of-speech, Case, phi-features, and dependency relation from its parent. We further tested ‘structural’ versions of the models in which the token’s parent index is also given as part of the stacked one-hot embedding, but found that this additional information had a negligible impact on model performance in most settings.

²These transformers are much smaller than state-of-the-art models. However, our preliminary tests with larger models showed drastic decreases in performance, likely due to the smaller size of our training data.

5.2 Datasets

5.2.1 Hindi UD Treebank

To evaluate our model on naturalistic data, we sourced trees from the Hindi Universal Dependencies Treebank (HDTB) (Bhat et al., 2017; Palmer et al., 2009), a manually annotated collection of sentences from news articles, heritage and tourism sites, and a small amount of conversational data. The standard split of this treebank contains 13,304 training sentences, 1,659 validation sentences, and 1,286 test sentences.

5.2.2 Synthetic Data

For more controlled data that includes the agreement phenomena of interest, we also wrote a probabilistic grammar that generates basic syntactic trees within the UD framework. This grammar allowed us to evaluate models without the annotation inconsistencies present in parts of HDTB, as well as to precisely control the types and frequencies of structures in the learning data. Specifically, we created production rules that generate transitive, intransitive, and ditransitive sentence frames in the perfective, progressive, and habitual aspects. Acceptable Case marking patterns are defined according to Hindi-Urdu’s split-ergativity (Keine, 2007; Mohanan, 1994; Butt, 1995). Verbs can either be simple predicates or light verb constructions, and can also introduce an embedded infinitival clause. To account for optionality, we introduce a flag on the infinitival clauses in which LDA is desired. Embedded infinitivals can also introduce an agreeing light verb construction as in (10). The full grammar can be found in the Appendix.

A *Full Set* of trees is generated by normalizing probability across each structure type. This contains 1700 sentences total, of which 1000 are used for training, 200 reserved for validation, and 500 are reserved for evaluation. We additionally generated a *Minimal Training Set* of examples by enumerating over all 98 structures possible from our grammar and then randomly permuting the number of auxiliaries and the phi-features on noun goals. This resulted in a set of 98 dependency trees. Finally, we created a *Relative Clause Test Set* by randomly appending relative clauses to 25% of the eligible goals in the original 500-sentence test set.

These sets were used in three tasks: a **Synthetic (Synth)** task that is trained, validated, and tested on the *Full Set*, a **Minimal** task that is trained on the *Minimal Training Set* but validated and tested

on the *Full Set*, and a **Relative Clause (ReCl)** task that is trained and validated on the *Full Set* but tested on the *Relative Clause Test Set*.

5.3 Results

The average test accuracies over 10 runs of each model are shown in Table 1. Each model was trained for a minimum of 1000 steps and a maximum of 100,000 steps, saving the checkpoint with the lowest validation loss for testing.

We find that the models performed similarly on the naturalistic treebank (HDTB). Our Search model slightly outperforms the transformer models without structural information, but not the Cloze model with access to parent information. Each model also performed similarly on the synthetic task, with both the Search model and the Cloze models reaching perfect or near-perfect test accuracy. However, compared to our Search model, the transformer models see a larger drop-off in synthetic accuracy in the low-data setting of the minimal task. This suggests that our Search model is particularly well-suited to low-resource data.

Most strikingly, while our Search model maintains near-perfect test accuracy on the relative clause generalization task, all of the baseline transformer models show a significant drop in performance compared to other tasks. This demonstrates an ability of our model to generalize agreement patterns to held-out examples and structures that the transformer models do not share. We hypothesize that the poor performance of the latter is due to an overreliance on heuristics— they have difficulty avoiding agreement with the subject and object distractors introduced by the relative clauses because they lack the structural biases of Search.

We note that the transformer models performed similarly with or without access to structural information (parent indexes), with the possible exception of the Cloze model on the naturalistic treebank. This suggests that these models do not consistently assign high weights to structural relations relative to other cues such as dependency relation or part of speech.

5.4 Learned Search Algorithm

To further examine the search algorithm that our model induces, we dissect a subset of a particular model’s learned weights (see Table 2). We can see that the model has learned a coherent search algorithm for Hindi-Urdu agreement. Weights on all phi-features are similar, suggesting that the model

		Gender Accuracies			Number Accuracies			Overall	
		Model	Masculine	Feminine	Total	Singular	Plural		Total
Dataset	HDTB	Search	0.904 ± 0.026	0.904 ± 0.012	0.904 ± 0.019	0.990 ± 0.003	0.796 ± 0.032	0.96 ± 0.005	0.924 ± 0.011
		Cloze	0.965 ± 0.004	0.808 ± 0.011	0.924 ± 0.003	0.978 ± 0.003	0.846 ± 0.017	0.958 ± 0.001	0.909 ± 0.002
		Cloze*	0.970 ± 0.006	0.867 ± 0.019	0.942 ± 0.004	0.987 ± 0.003	0.826 ± 0.013	0.963 ± 0.002	0.942 ± 0.003
		LM	0.940 ± 0.009	0.782 ± 0.033	0.898 ± 0.006	0.975 ± 0.003	0.778 ± 0.02	0.945 ± 0.002	0.881 ± 0.005
	LM*	0.947 ± 0.012	0.778 ± 0.028	0.902 ± 0.004	0.977 ± 0.004	0.785 ± 0.020	0.947 ± 0.002	0.888 ± 0.003	
	Synth	Search	1.0 ± 0	1.0 ± 0	1.0 ± 0	1.0 ± 0	1.0 ± 0	1.0 ± 0	1.0 ± 0
		Cloze	1.0 ± 0	0.999 ± 0.0008	0.999 ± 0.0003	1.0 ± 0	1.0 ± 0	1.0 ± 0	0.999 ± 0.0003
		Cloze*	1.0 ± 0	1.0 ± 0	1.0 ± 0	1.0 ± 0	1.0 ± 0	1.0 ± 0	1.0 ± 0
		LM	0.991 ± 0.005	0.992 ± 0.001	0.991 ± 0.003	0.984 ± 0.009	0.995 ± 0.001	0.989 ± 0.005	0.983 ± 0.007
	LM*	0.992 ± 0.005	0.992 ± 0.000	0.992 ± 0.003	0.989 ± 0.007	0.989 ± 0.008	0.989 ± 0.006	0.982 ± 0.008	
	Minimal	Search	0.995 ± 0.01	0.995 ± 0.014	0.995 ± 0.011	0.99 ± 0.027	0.996 ± 0.014	0.993 ± 0.02	0.989 ± 0.029
		Cloze	0.990 ± 0.004	0.969 ± 0.015	0.982 ± 0.007	0.979 ± 0.002	0.995 ± 0.009	0.986 ± 0.004	0.972 ± 0.007
		Cloze*	0.989 ± 0.004	0.951 ± 0.017	0.973 ± 0.005	0.980 ± 0.002	0.977 ± 0.018	0.978 ± 0.007	0.960 ± 0.007
		LM	0.987 ± 0.003	0.906 ± 0.073	0.954 ± 0.031	0.943 ± 0.094	0.899 ± 0.165	0.924 ± 0.125	0.896 ± 0.131
	LM*	0.982 ± 0.026	0.846 ± 0.109	0.927 ± 0.059	0.876 ± 0.198	0.853 ± 0.212	0.866 ± 0.156	0.812 ± 0.180	
	ReCI	Search	0.996 ± 0.008	1.0 ± 0	0.998 ± 0.005	0.995 ± 0.010	1.0 ± 0	0.997 ± 0.006	0.997 ± 0.005
		Cloze	0.828 ± 0.013	0.904 ± 0.013	0.861 ± 0.004	0.833 ± 0.013	0.915 ± 0.016	0.870 ± 0.001	0.797 ± 0.004
		Cloze*	0.829 ± 0.026	0.890 ± 0.025	0.855 ± 0.004	0.852 ± 0.008	0.876 ± 0.014	0.863 ± 0.004	0.787 ± 0.011
		LM	0.846 ± 0.016	0.894 ± 0.009	0.867 ± 0.006	0.820 ± 0.041	0.928 ± 0.018	0.869 ± 0.015	0.802 ± 0.02
	LM*	0.828 ± 0.035	0.866 ± 0.022	0.844 ± 0.013	0.833 ± 0.015	0.878 ± 0.023	0.853 ± 0.008	0.774 ± 0.02	

Table 1: Test accuracies for each model broken down by phi-feature type and value, where * indicates that a transformer model had access to structural information about node parents.

does not prioritize any particular phi-feature combination (e.g., masculine singular) over others. Taking the weights on Case and dependency relation together, we see that the model strongly prefers Nominative subjects, and prefers Nominative objects over Ergative subjects. Moreover, the default weight by itself is preferred over an Ergative subject and an Accusative object. To additionally handle LDA and light verb agreement, we see a very high weight on embedded infinitival clauses, likely to overcome the otherwise low priority given to verbs. On the other hand, low priority is given to light verb noun compound dependents, as Nominative nouns are already given high priority

In practice, the learned weights of Search encourage the softmax that the model takes at each time step to be close to one-hot. Thus, by examining the softmax scores at each time step, we can recover the ‘path’ that a probe takes to reach its goal. We sketch one such path in Figure 1. In this example of long-distance agreement, both probes must take multiple steps to reach their goal. The verb probe must first take the compound transition to its embedded infinitival clause, from where it can then transition to the embedded object. The tense probe requires an additional iteration, first taking the auxiliary arc to the root verb, then the compound arc to the infinitival verb, and then finally the object arc to the embedded object. The model has learned a coherent and efficient search path from each probe to the correct goal.

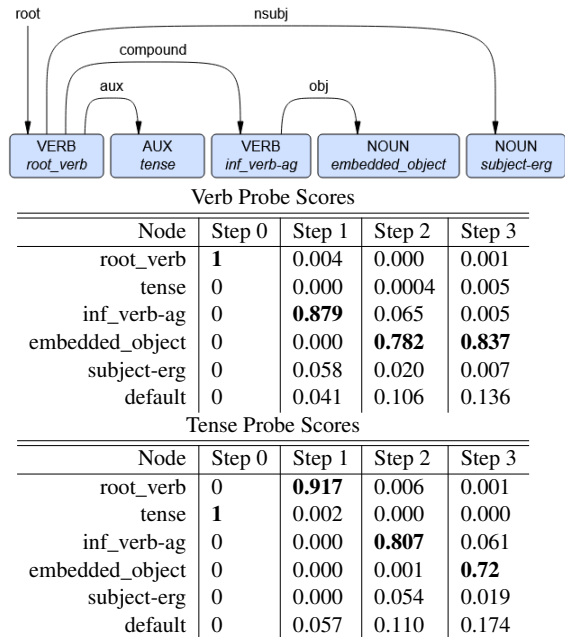


Figure 1: Attention patterns at each step for the verb and tense probe for a sentence with long distance agreement.

Case	Weight	Phi-Features	Weight	Part of Speech	Weight	Dependencies	Weight
Nominative	7.96	Masculine	2.79	Noun	3.33	Subject dependent	4.36
Accusative	-4.55	Feminine	3.04	Verb	0.003	Object dependent	-4.79
Ergative	-6.58	Singular	2.77	Auxiliary	-1.74	Infinitival Clause dependent	6.40
		Plural	2.85			Light Verb Noun Compound dependent	0.61
						Auxiliary head	10.11
						Default node	3.49

Table 2: A subset of learned weights for a model trained on synthetic data. Taken together, we see that the model prefers Nominative (unmarked) subjects over all objects, Nominative (unmarked) objects over Ergative subjects, the default dummy node over Ergative subjects and Accusative objects. We also see a high preference for embedded infinitival clauses (6.40) to overcome the otherwise low preference for verbs (0.003), and a high preference for the heads of auxiliaries (10.11) to allow auxiliary probes to travel to the matrix verb.

6 Conclusion and Future Directions

Artificial neural networks are often seen as black-box models with little or no inductive bias. We present a counterpoint to this view, creating an efficient, minimal, and interpretable neural network model that possesses a strong inductive bias for agreement as structurally-informed search.

Our goal in building this model is not necessarily to adjudicate between neural networks and traditional symbolic models as opposing models of language or cognition. Rather, we aim to show that insights from symbolic modeling can provide useful inductive biases for neural network models. Indeed, our structure-dependent model is capable of correctly learning a search algorithm for the agreement pattern in Hindi-Urdu, and matches or exceeds performance compared to much larger models without such biases. Our model is also capable of achieving near-perfect performance on a structural generalization task, something that more generic models could not match.

While we tested our model on the complex agreement system of Hindi-Urdu, our model is theoretically capable of accounting for a range of agreement phenomena cross-linguistically. For example, an agreement system in which a verb obligatorily agrees with the subject of a clause can be easily accounted for by setting a high weight on the *subject* dependency (nsubj). Our model can also capture the various sensitivities that agreement has with Case in languages other than Hindi-Urdu. Nepali, for example, allows agreement with Ergative subjects as well as Nominative subjects, while Gujarati allows agreement with Accusative objects but not Ergative subjects (Bhatt, 2005). Our model can capture the Nepali case with an equal setting of our Case weights for Nominative and Ergative, and the Gujarati case with a positive weight on Accusative and a negative weighting of Ergative.

However, there do exist some agreement phenomena that our model cannot yet account for. Our model is specified to return a simple weighted combination of phi-features among existing nodes in a tree, making it impossible to account for agreement with coordinated noun phrases that have phi-features computed by ‘resolution rules’ applied to their constituents (Bhatia, 2011). Additionally, the weighted combination that our model returns is often exactly the phi-features from a single node, as the model typically converges to near one-hot attention patterns after training. Thus, it seems unlikely that the model can account for agreement phenomena that depend on multiple goals (Shen, 2019) — though distribution of attention over multiple nodes does remain a logical possibility and may be encouraged by some training patterns.

Finally, the model as deployed here does not provide a perfect match to the theories of agreement typically proposed by syntacticians. While most theoretical work on agreement is oriented around constituency trees, our model was trained and tested on dependency trees. However, the model can be minimally adapted to operate on any tree structure, including constituency trees, giving us the potential to address questions regarding directionality and feature weighting in other settings.

Acknowledgements

Thanks to Paul Smolensky, members of the PhonMorph and the Neurosymbolic Computation labs at Johns Hopkins University, and two anonymous SCiL reviewers for useful comments and questions. We would also like to thank Rajesh Bhatt for his insights on Hindi-Urdu agreement and syntax. This research was partially supported by NSF grant BCS-1941593 to CW.

References

- Tafseer Ahmed, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2012. [A reference dependency bank for analyzing complex predicates](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, page 3145–3152, Istanbul, Turkey. European Language Resources Association (ELRA).
- Geoff Bacon and Terry Regier. 2019. [Does BERT agree? Evaluating knowledge of structure dependence through agreement relations](#). (arXiv:1908.09892). ArXiv:1908.09892 [cs].
- Mark C. Baker. 2008. *The Syntax of Agreement and Concord*. Cambridge University Press, Cambridge.
- Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. 2017. The Hindi/Urdu treebank project. In *Handbook of Linguistic Annotation*. Springer Press.
- Archna Bhatia. 2011. *Agreement in the context of coordination Hindi as a case study*. Ph.D. thesis, University of Illinois at Urbana-Champaign.
- Rajesh Bhatt. 2005. [Long distance agreement in Hindi-Urdu](#). *Natural Language & Linguistic Theory*, 23(4):757–807.
- Rajesh Bhatt and Stefan Keine. 2017. [Long-Distance Agreement](#). In Martin Everaert and Henk C. van Riemsdijk, editors, *The Wiley Blackwell Companion to Syntax, Second Edition*, pages 1–30. John Wiley & Sons, Inc., Hoboken, NJ.
- Bronwyn M. Bjorkman and Hedde Zeijlstra. 2019. [Checking up on Agree](#). *Linguistic Inquiry*, 50(3):527–569.
- Jonathan David Bobaljik. 2008. Where’s phi? Agreement as a post-syntactic operation. In Daniel Harbour, David Adger, and Susana Béjar, editors, *Phi-Theory: Phi features across interfaces and modules*, pages 295–328.
- Miriam Butt. 1993. [A reanalysis of long distance agreement in Urdu](#). In B.Kaiser and C. Zoll, editors, *Proceedings of the Nineteenth Annual Meeting of the Berkeley Linguistics Society: General Session and Parasession on Semantic Typology and Semantic Universals (1993)*, volume 19, page 52–63.
- Miriam Butt. 1995. *The structure of complex predicates in Urdu*. CSLI Publications, Stanford, CA.
- Susana Béjar and Milan Rezac. 2009. [Cyclic agree](#). *Linguistic Inquiry*, 40(1):35–73.
- Aditi Chaudhary, Antonios Anastasopoulos, Adithya Pratapa, David R. Mortensen, Zaid Sheikh, Yulia Tsvetkov, and Graham Neubig. 2020. [Automatic extraction of rules governing morphological agreement](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5212–5236, Online. Association for Computational Linguistics.
- Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.
- Greville G Corbett. 2006. *Agreement*. Cambridge University Press, Cambridge.
- Amy Rose Deal. 2015. [Interaction and satisfaction in phi-agreement](#). In Thuy Bui and Deniz Ozyildiz, *Proceedings of NELS 45*, Volume 1, page 179–192. Amherst: GLSA.
- Yoav Goldberg. 2019. [Assessing BERT’s syntactic abilities](#). *arXiv*.
- Yamuna Kachru. 1970. [An introduction to Hindi syntax](#). *Journal of Linguistics*, 6(1):151–152.
- Alan Hezao Ke. 2023. [Can Agree and Labeling be reduced to Minimal Search?](#) *Linguistic Inquiry*, pages 1–22.
- Stefan Keine. 2007. Reanalysing Hindi split-ergativity as a morphological phenomenon. *Linguistische Arbeits Berichte*, 85:73–127.
- Bingzhi Li, Guillaume Wisniewski, and Benoît Crabbé. 2023. [Assessing the capacity of transformer to abstract syntactic representations: A contrastive analysis based on long-distance agreement](#). *Transactions of the Association for Computational Linguistics*, 11:18–33.
- 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.
- Anoop Kumar Mahajan. 1990. *The A/A-bar distinction and movement theory*. Ph.D. thesis, Massachusetts Institute of Technology.
- Tara Mohanan. 1994. *Argument structure in Hindi*. CSLI Publications, Stanford, CA.
- Edith A. Moravcsik. 1978. Agreement. In Charles A. Ferguson & Edith A. Moravcsik Joseph H. Greenberg, editor, *Universals of Human Language. Vol. IV: Syntax*, page 331–374. Stanford University Press, Stanford, CA.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. [Universal Dependencies v2: An evergrowing multilingual treebank collection](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.
- Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure.

- In *The 7th International Conference on Natural Language Processing*, pages 14–17.
- Rajeshwari Pandharipande and Yamuna Kachru. 1977. [Relational grammar, ergativity, and Hindi-Urdu](#). *Lingua*, 41(3):217–238.
- Omer Preminger. to appear. [Phi-feature agreement in syntax](#). In Kleanthes K. Grohmann and Evelina Leivada, editors, *The Cambridge Handbook of Minimalism*. Cambridge University Press, Cambridge.
- Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. 2018. [Can LSTM learn to capture agreement? The case of Basque](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, page 98–107, Brussels, Belgium. Association for Computational Linguistics.
- Zheng Shen. 2019. [The multi-valuation agreement hierarchy](#). *Glossa: a journal of general linguistics*, 4(11).
- Ashwini Vaidya, Sumeet Agarwal, and Martha Palmer. 2016. [Linguistic features for Hindi light verb construction identification](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1320–1329, Osaka, Japan. The COLING 2016 Organizing Committee.
- Ashwini Vaidya, Owen Rambow, and Martha Palmer. 2019. [Syntactic composition and selectional preferences in Hindi Light Verb Constructions](#). *Linguistic Issues in Language Technology*, 17.
- Michael Wilson, Zhenghao Zhou, and Robert Frank. 2023. [Subject-verb agreement with seq2seq transformers: Bigger is better, but still not best](#). *Society for Computation in Linguistics*, 6:278–288.

A Synthetic grammar

Our synthetic grammar, designed to capture the agreement phenomena of interest in the paper, is shown below. Each row corresponds to an expansion rule of the grammar. The leftmost number of each row corresponds to the weight of that expansion rule, while the first entry immediately after the number corresponds to the parent node that the expansion rule targets. The remaining entries are nodes that are added to the tree as children of the parent node. Entries with parentheses are optional and generated with 50% probability. For example, the rule 1.35 root_verb subject-erg object-nom (tense) denotes a rule with weight 1.35 that expands a root_verb node with an Ergative subject child, an Nominative object child, and an optional tense child. In practice, each node is fully specified for features, dependency relation, and part of speech, but this has been truncated here for readability.

```
# ROOT
2 R root_verb
1 R root_verb_prog

# HABITUAL AND PERFECTIVE
# Simple Transitive
1.35 root_verb subject-erg object-nom (tense)
1.35 root_verb subject-nom object-nom (tense)
1.35 root_verb subject-nom object-acc (tense)
1.35 root_verb subject-erg object-acc (tense)
# Simple Intransitive
2.7 root_verb subject-erg (tense)
2.7 root_verb subject-nom (tense)
# Simple Ditransitive
2.7 root_verb subject-erg object-dat object-nom (tense)
2.7 root_verb subject-nom object-dat object-nom (tense)
# Light Verb Constructions
0.385 root_verb subject-nom object-nom host_adj (tense)
0.385 root_verb subject-nom object-acc host_adj (tense)
0.385 root_verb subject-nom object-nom host_verb (tense)
0.385 root_verb subject-nom object-acc host_verb (tense)
0.385 root_verb subject-nom object-nom host_noun (tense)
0.385 root_verb subject-nom object-acc host_noun (tense)
0.385 root_verb subject-nom host_noun_agreeing (tense)
0.385 root_verb subject-erg object-nom host_adj (tense)
0.385 root_verb subject-erg object-nom host_verb (tense)
0.385 root_verb subject-erg object-nom host_noun (tense)
0.385 root_verb subject-erg host_noun_agreeing (tense)
0.385 root_verb subject-erg object-acc host_adj (tense)
0.385 root_verb subject-erg object-acc host_verb (tense)
0.385 root_verb subject-erg object-acc host_noun (tense)
# Infinitivals
1.08 root_verb subject-erg inf_verb-agree (tense)
1.08 root_verb subject-nom inf_verb-nonagree (tense)
1.08 root_verb subject-nom inf_verb-nonagree-acc (tense)
1.08 root_verb subject-erg inf_verb-nonagree (tense)
1.08 root_verb subject-erg inf_verb-nonagree-acc (tense)

# PROGRESSIVE
# Simple Transitive
1.2 root_verb_prog subject-nom object-nom aspect (tense)
1.2 root_verb_prog subject-nom object-acc aspect (tense)
# Simple Intransitive
2.4 root_verb_prog subject-nom aspect (tense)
# Simple Ditransitive
2.4 root_verb_prog subject-nom object-dat object-nom aspect (tense)
# Light Verb Constructions
0.34 root_verb_prog subject-nom object-nom host_adj aspect (tense)
0.34 root_verb_prog subject-nom object-acc host_adj aspect (tense)
0.34 root_verb_prog subject-nom object-nom host_verb aspect (tense)
0.34 root_verb_prog subject-nom object-acc host_verb aspect (tense)
0.34 root_verb_prog subject-nom object-nom host_noun aspect (tense)
0.34 root_verb_prog subject-nom object-acc host_noun aspect (tense)
0.34 root_verb_prog subject-nom host_noun-agreeing aspect (tense)
# Infinitivals = 1
2.4 root_verb_prog subject-nom inf_verb-nonagree aspect (tense)

# EXPANSIONS
# Light Verb Construction Expansions
```

```
1 host_agreeing object-gen
1 host_agreeing object-loc
1 host_agreeing object-ins
1 host_agreeing

# Agreeing Infinitival Expansions
1 inf_verb-agreeing object-nom
1 inf_verb-agreeing host_noun-agreeing

# Non-Agreeing Infinitival Clause Expansions
1 inf_verb-non object-nom
1 inf_verb-non object-acc
```


Interference Predicts Locality: Evidence from an SOV language

Sidharth Ranjan
University of Stuttgart
sidharth.ranjan03@gmail.com

Sumeet Agarwal
IIT Delhi
sumeet@iitd.ac.in

Rajakrishnan Rajkumar
IIIT Hyderabad
raja@iiit.ac.in

Abstract

LOCALITY and INTERFERENCE are two mechanisms which are attested to drive sentence comprehension. However, the relationship between them remains unclear—are they alternative explanations or do they operate independently? To answer this question, we test the hypothesis that in Hindi, interference effects (measured by semantic similarity and case markers) significantly predict locality effects (modelled using dependency length quantifying distance between syntactic heads and their dependents) within a sentence, while controlling for expectation-based measures and discourse givenness. Using data from the Hindi-Urdu Treebank corpus (HUTB), we validate the stated hypothesis. We demonstrate that sentences with longer dependency length consistently have semantically similar preverbal dependents, more case markers, greater syntactic surprisal, and violate intra-sentential givenness considerations. Overall, our findings point towards the conclusion that locality effects are reducible to broader memory interference effects rather than being distinct manifestations of locality in syntax. Finally, we discuss the implications of our findings for the theories of interference in comprehension.

1 Introduction

The language comprehension system is long known to be constrained by working memory considerations (Ebbinghaus, 1885; Yngve, 1960; Ebbinghaus, 2013). Several theories and mechanisms of *syntactic complexity* have been proposed in sentence comprehension literature to account for processing difficulties. LOCALITY and INTERFERENCE are two such mechanisms by which online, incremental processing happens in language comprehension (Vasishth, 2011). As defined in that work, locality is the claim that the distance between syntactically related words (*i.e.*, dependent and head) determines the difficulty of integrating

a dependent with its head in a syntactic structure, owing to limited working memory capacity during comprehension. In contrast, the notion of interference denotes situation wherein linguistic elements sharing common characteristics, such as form, meaning, animacy, or concreteness, result in processing difficulties when they are situated nearby or in close proximity (Lewis, 1996). Notably, interference effects causing forgetting has a long history in cognitive psychology literature, suggesting that interference could be a manifestation of memory overload during retrieval (Baddeley and Hitch, 1977; Roediger III and Abel, 2022).

In an extensive survey of locality and interference effects and their interplay, Vasishth (2011) proposed that locality and interference may represent two sides of the same underlying memory effects. For example, locality represented in terms of the number of discourse referents between head and dependent in theories like Dependency Locality Theory (DLT, Gibson, 2000), primarily reflecting the accessibility or availability of intervening elements. Conversely, interference emerges from the similarity among intervening materials, thereby impeding the dependent’s integration at the head. Therefore, it is not clear yet if locality and interference are two alternative explanations or do both the factors operate independently? Vasishth characterizes this point as the *locality-interference* debate and advocates an empirical investigation to disentangle their relative impact.

Our work addresses this gap in the literature by investigating the relationship between locality and interference effects using observational data from naturally occurring sentences in Hindi. We test the hypothesis that interference effects significantly predict dependency length within a sentence, while we statistically control for predictability measures and givenness discourse considerations as potential confounds. We quantify *locality effects* using dependency length, which is inspired from integra-

tion costs posited by DLT, and *interference effects* using semantic similarity among preverbal heads and metrics based on case density (number of case markers per sentence). Hindi, an Indo-Aryan language within the Indo-European family, exhibits a robust case-marking system and flexible word order with Subject-Object-Verb (SOV) as its canonical structure (Kachru, 2006):

- (1) amar ujala-ko yah sukravar-ko daak-se
 Amar Ujala-ACC it friday-on post-INST
 prapt hua
 receive be.PST.SG

Amar Ujala received it by post on Friday.

Our hypothesis is inspired by a substantial body of evidence from the studies on dependency locality and interference effects across languages (Staub, 2010; Vasishth, 2011; Vasishth and Drenhaus, 2011; Jäger et al., 2015; Ranjan et al., 2019; Stone et al., 2020). These studies suggest that language processor concurrently exhibits both dependency distance and interference minimization to overcome pressure related to working memory load. In contrast to English, verb-final languages like Hindi lack strong empirical support for locality effects (Vasishth and Lewis, 2006; Husain et al., 2014, 2015; Ranjan et al., 2022a,b; Ranjan and van Schijndel, 2024, cf. Ranjan and von der Malsburg, 2023; Ranjan and von der Malsburg, 2024) and numerous instances of anti-locality effects have been reported. Levy (2008) demonstrated that anti-locality patterns can be effectively explained using expectation-based accounts such as surprisal theory, with a view that introduction of more intervening words sharpens expectations at the verbal integration site within the sentence, thereby aiding comprehension. However, Vasishth and Lewis (2006) proposed an alternative unified explanation that explains both locality and anti-locality effects in Hindi. Based on Adaptive Control of Thought—Rational (ACT-R) framework (Anderson and Paulson, 1977), Vasishth and colleagues suggest that these effects can either be on account of activation decay in memory (anti-locality) or due to interference of intervening elements (locality). Subsequent studies in the literature advocate for a comprehensive theory of syntactic complexity encompassing both expectation-based and memory-based theories (Levy et al., 2013; Husain et al., 2014; Ranjan et al., 2022b).

To test the stated hypothesis, we deploy data from Hindi-Urdu Treebank (Bhatt et al., 2009,

HUTB) corpus containing written text from newswire domain. We compute sentence-level cognitive measures: dependency length as *locality* measure, trigram surprisal and PCFG surprisal as *predictability* measures, semantic similarity and case-marker features as *interference* measures, and lastly, information status as *discourse* measure. We then compute their averaged values throughout each sentence and subsequently, fit a linear regression model to predict the average dependency length of corpus sentences. This approach is well motivated from the previous studies that have tried explaining dependency locality in natural languages (Futrell, 2019; Sharma et al., 2020). Our results demonstrate that similarity-based interference (as modelled using semantic similarity of preverbal dependents) is a significant predictor of dependency length for the entire dataset as well as for specific constructions of interest, *viz.*, non-canonical OSV orders and conjunct verbs. Our analysis of different bins of increasing dependency lengths consistently revealed higher occurrences of case markers and semantically similar elements. Overall, our findings suggest that dependency length, indicative of locality effects, is modulated by more general memory interference effects. Finally, we discuss the implications of our findings for the theories of interference in comprehension.

Our main contribution is that we provide an empirical basis for the Vasishth (2011)’s theoretical proposal on the *interference-locality* debate using broad-coverage study in Hindi. Moreover, we make use of naturally occurring sentences as opposed to artificially crafted sentences in a controlled laboratory experiments, thereby providing broader significance to the presented findings. Finally, our work extends the scope of psycholinguistic research beyond an anglocentric focus, allowing for a broader typological base for theory development (Jaeger and Norcliffe, 2009; Norcliffe et al., 2015).

2 Measures of Processing Difficulty

In this section, we present details of the theories alluded to in the introduction and measures derived from them for our experiments.

2.1 Locality measure

Dependency Locality Theory (Gibson, 1998, 2000, DLT) has been successfully shown to empirically predict the source of comprehension difficulty within sentences. DLT quantifies memory load

during sentence comprehension in two ways: a) by counting the number of new discourse referents introduced between heads and dependents (INTEGRATION COST), b) by counting the number of incomplete dependencies (upcoming heads) at a given word that needs to be stored in memory (STORAGE COST). The theory assumes *decay* as its underlying cognitive construct, suggesting that as the distance between head-dependent pairs increases, the information fades away, leading to *forgetting*. As a result, language users strive to minimize the distance between syntactically related units in the sentence. The aforementioned DLT metrics have been successfully shown to account for greater complexity (measured in terms of reading times) of object relative clauses compared to subject relatives in English, and more generally have been influential in shaping natural languages (Liu, 2008; Futrell et al., 2015, 2020; Ranjan and von der Malsburg, 2023, 2024).

Inspired by Gibson’s word-by-word integration cost, we define the dependency length as the count of intervening words between head and dependent units within a dependency graph (Temperley, 2007). Figure 4 in Appendix A illustrates the calculation of dependency length for Example sentence 1. The average dependency length for each sentence was computed by dividing the total dependency length (sum of per-word dependency lengths) by the sentence length (count of words in the sentence).

2.2 Interference measure

Previous studies on *similarity-based interference* have quantified the comprehension difficulty by examining the intervening materials between syntactically related units, and the elements retrieved at the integration site in the sentence (Gordon et al., 2006, 2002, 2001; Van Dyke and Lewis, 2003; Van Dyke and McElree, 2006; Jager et al., 2017; Lewis, 1998; Lee et al., 2005; Van Dyke and Johns, 2012). These studies reported that comprehender tends to make more retrieval errors when they experience similar items that need to be retrieved from the working memory. This happens because similar items share common feature attributes in the memory and cause undesired confusion while retrieving the correct target element. For instance, both Traxler et al. (2002) and Staub (2010) contend that the processing difficulties associated with English relative clause examples shown in Figure 1 can be explained using interference effects in addition to distance effects. Sentences containing

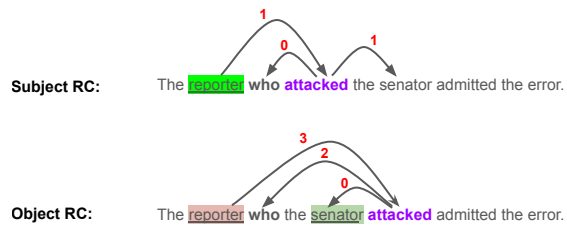


Figure 1: Interference-locality debate

objective relative clauses (ORC) are typically more challenging to process and comprehend than subject relative clauses (SRC) due to their greater dependency length. However, this behavior can also be attributed to interference phenomena. The ORC structures involve more interference compared to SRC structures. The nouns like ‘reporter’ and ‘senator’ in ORCs, both falling into the same category, induce greater interference during retrieval at the inner verb ‘attacked,’ unlike their SRC counterparts.

As pointed previously, the interference explanation has its independent motivation from the theory of working memory retrieval—*cue-based retrieval model* derived from the ACT-R cognitive architecture, which includes *decay*, *re-activation*, *cue-matching*, and *interference* (Lewis and Vasishth, 2005; Anderson et al., 2004). Contrasting DLT’s decay and retrieval interference mechanisms, Vasishth (2011) expounds that decay is a lack of focused attention over *to-be-retrieved* information when the processor is engaged with interpreting the intervening elements. On the contrary, interference is about attention being shared unnecessarily to multiple units of information leading to *unavailability* of the required information. Therefore, under this logical space, Vasishth posits that DLT and interference could be the two manifestations of the same phenomenon that we intend to probe in the current work.

We operationalize these insights by estimating the semantic similarity between adjacent preverbal heads (directly linked to the main-verb) in the sentence. The similarity scores $sim(d_i)$ for each preverbal head was estimated by computing the cosine similarity (Salton, 1972) of the target head word with the adjacent head-word (see Equation 1). For instance, consider the sentence Example 1 and corresponding dependency graph shown in Figure 2, we computed the cosine similarity between following pairs: (ujala, yah); (yah, sukraavar); (sukraavar, daak); (daak, prapt).

$$\text{sim}(d_i) = \frac{wv(d_i) \cdot wv(d_{i+1})}{\|wv(d_i)\| \|wv(d_{i+1})\|} \quad (1)$$

$wv(d)$ and $wv(d_{i+1})$ denote the word vectors of the head-word d_i and d_{i+1} , respectively. These word vectors were obtained from the pre-trained *word2vec* model for Hindi (Grave et al., 2018). We then calculated the average semantic similarity by summing the similarity score over all the preverbal heads and then divided it by total preverbal heads in the sentence. As a sanity check, we deployed this method to understand how well the cosine similarity predicts the human judgment ratings¹ of word-similarity in Hindi (Bhatia et al., 2021). Notably, we found that the cosine similarity had a Spearman’s rank correlation of 0.75 with the human judgment ratings, signifying its capability to model interference effects. Prior work has shown that cosine similarity metric is effective in modeling interference phenomenon (Sharma et al., 2020; Smith and Vasishth, 2020) and reading time (Frank, 2017; Salicchi et al., 2021) but also see Merlo and Ackermann (2018) and De Deyne et al. (2016).

2.3 Predictability measures

Surprisal theory (Hale, 2001; Levy, 2008) posits that language knowledge (or grammar) is probabilistic in nature, shaped by prior linguistic experiences and language learning. The cited authors suggest that the cognitive effort required to comprehend a word w_k in its context can be quantified using Shannon’s information-theoretic measure of the log of the inverse of word’s conditional probability given the preceding context ($w_{1...k-1}$):

$$\text{Effort}(w_k) \propto \log \frac{1}{P(w_k|w_{1...k-1})} \quad (2)$$

$$S_k = -\log P(w_k|w_{1...k-1}) = \log \frac{P(w_{1...k-1})}{P(w_{1...k})} \quad (3)$$

Subsequently, the *surprisal* of the k^{th} word, w_k , is defined as the negative log probability of w_k given the preceding intra-sentential ($w_{1...k-1}$) context (see Equation 3). These probabilities can be computed either over word sequences or syntactic configurations and reflect the information load (or predictability) of w_k . The theory is supported by a large body of empirical evidence from behavioural as well as broad-coverage corpus data (Demberg and Keller, 2008; Boston et al., 2008; Roark et al.,

¹https://github.com/ashwinivd/similarity_hindi

2009; Agrawal et al., 2017; Dammalapati et al., 2021; Ranjan et al., 2022a). The cited studies suggest that words with high surprisal tend to have high reading time. In this work, we estimated the per-word lexical trigram surprisal using n-gram language model and syntactic surprisal using probabilistic context-free grammar (PCFG) parser as described below.

- **Trigram surprisal:** We estimated the lexical surprisal for each word in the sentence using a 3-gram language model (LM) trained on the written section of the EMILLE Hindi Corpus (Baker et al., 2002), consisting of 1 million sentences, using the SRILM toolkit (Stolcke, 2002) with Good-Turing discounting.
- **PCFG surprisal:** We estimated the syntactic surprisal of each word in a sentence using the Berkeley latent-variable PCFG parser² (Petrov et al., 2006). We trained the parser using 12000 phrase structure trees obtained by converting Bhatt et al.’s HUTB dependency trees into constituency trees using the approach described in Yadav et al. (2017). We adopted the 5-fold cross-validation approach to compute the surprisal scores from the PCFG parser i.e., surprisal for each test sentence was estimated by training a PCFG LM on four folds of the phrase structure trees and then testing on a fifth held-out fold.

For both measures above, we computed the average surprisal for each sentence by summing the word-level surprisal of all the words in the sentence and then divided it by sentence length.

2.4 Information status measure

Languages are known to obey *given-before-new* principle by producing elements that are previously mentioned in the discourse prior to the new content in the sentence (Clark and Haviland, 1977; Chafe, 1976; Kaiser and Trueswell, 2004).

In this work, we annotate each sentence in our dataset with GIVEN-NEW ordering. The preverbal subject and object constituents of a target sentence were assigned *Given* tag if any content word within the phrase appeared in the preceding sentence in the corpus or the head of the phrase was a pronoun; else, the phrases were tagged as *New*. We then assigned scores to each ordering as per the following

²5-fold cross-validated parser training and testing F1-score metrics were 90.82% and 84.95%, respectively.

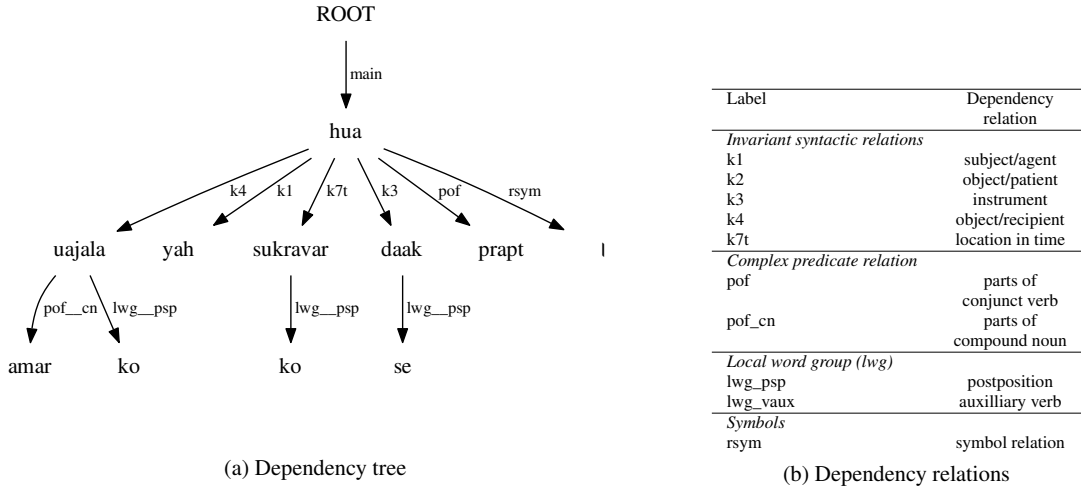


Figure 2: HUTB dependency tree and corresponding dependency relation labels for Example 1

scheme: a) New-Given = -1 b) Given-New = +1 c) Given-Given and New-New = 0. See Appendix B for an illustration.

2.5 Case markers

In Hindi, case markers are identified as postpositions,³ crucial for conveying grammatical relationships within sentences (Kachru, 2006; Agnihotri, 2007). Case markers influence comprehension via the mechanisms of either predictability (Avetisyan et al., 2020) or interference effects (Tily, 2010; Ranjan et al., 2019). They have been shown to predict the upcoming verb (Husain et al., 2014; Grissom II et al., 2016) and effectively reduce interference in memory by correctly distinguishing between subject and objects, thereby enhancing retrieval at the verb. Moreover, there is extensive research on case marker interference in sentence comprehension in SOV languages like Japanese (Lewis and Nakayama, 2001), Korean (Lee et al., 2005) and Hindi (Vasishth, 2003). Inspired by these insights, we compute following measures to quantify the distinct effects of case markers:

- **Case density:** Ratio of the number of case markers to the word counts in the sentence.
- **Same case bigrams:** Total number of identical case marker sequences associated with pairs of adjacent preverbal constituents.

³Table 4 in Appendix C outlines Hindi case markers and their functions.

3 Data and Methods

Our dataset consists of 1996 declarative sentences with well-defined subject and object phrases from the Hindi-Urdu Treebank (HUTB) corpus of the written text belonging to the newswire domain (Bhatt et al., 2009). For each sentence, we first compute average values of various cognitive measures. We then fit a linear regression model to predict the average dependency length of sentences in our dataset, using the remaining cognitive measures discussed in the preceding section as independent variables. All the independent variables were normalized to z -scores, *i.e.*, the predictor’s value (centered around its mean) was divided by its standard deviation. We used the `glm` function in R to perform our regression experiments expressed using the `glm` equation below:

$$\text{Dependency length} \sim \left\{ \begin{array}{l} \text{similarity} + \text{same case bigram} + \\ \text{trigram surprisal} + \text{PCFG surprisal} + \\ \text{case density} + \text{IS score} \end{array} \right. \quad (4)$$

The pairwise Pearson correlation coefficients among these measures are shown in Appendix D, Figure 5. We observed a moderate correlation of 0.31 between dependency length and semantic similarity, whereas the remaining predictors exhibited weaker correlations with dependency length.

4 Results

In this section, we test the hypothesis that locality effects as captured by dependency length are

Predictor	$\hat{\beta}$	$\hat{\sigma}$	t
Intercept	1.87	0.016	118.81
IS Score	-0.04	0.013	-3.13
PCFG surprisal	0.06	0.019	2.99
3.gram surprisal	-0.05	0.019	-2.53
same case bigram	-0.01	0.015	-0.27
case density	0.09	0.015	5.94
similarity	0.19	0.015	13.16
PCFG x 3.gram surp	-0.05	0.011	-4.08

Table 1: Linear regression model predicting average dependency length on full data set (1996); significant predictors denoted in bold

reducible to more general memory-interference effects as captured by semantic similarity and case marker features while controlling for expectation-based measures and discourse givenness. We, therefore, expect that interference-based features should have positive regression coefficients in the regression model. In other words, sentences with longer dependency length in Hindi should exhibit greater interference effects as quantified by more case markers, and interfering noun phrases (NPs) with similar featural attributes. Our results are discussed in the remaining subsections.

4.1 Predicting dependency length

We first performed regression analyses on the entire data set to investigate the influence of predictability, interference, and givenness measures on the dependency length. We then reported the statistical analyses on different bins of dependency lengths. Table 1 displays the regression results over the entire data set. All interference measures other than same-case bigram counts are significant predictors of dependency length, thus validating our proposed hypothesis. The positive regression coefficient for semantic similarity indicates that with every unit increase in its score, the value of dependency length also increases, thus shedding light on how locality effects are modulated in Hindi. Moreover, adding similarity score into a model containing all other predictors significantly improved the fit of our regression model ($\chi^2 = 166.75$; $p < 0.001$). The positive regression coefficient of case density suggests that sentences with more case markers tend to have higher dependency length, consequently highlighting both predictability and interference effects of case markers as discussed in the comprehension literature (Husain et al., 2014; Avetisyan et al., 2020).

The negative regression coefficient of the IS score suggests that sentences with longer dependency length have NEW-GIVEN ordering. Finally, syntactic PCFG and lexical trigram surprisal measures have positive and negative regression coefficients, respectively, with a significant interaction between the two while predicting dependency length, suggesting that syntactically complex sentences may have more probable word sequences.

We investigated the relationship between case marking and dependency length in more detail. For each of the 25 most frequent verbs in the HUTB, we plotted the average case density of all sentences having that verb as the root of the sentence against the average dependency length of those sentences (refer to Figure 3). Many of the high-frequency verbs have an average dependency length greater than the average value for all verbs. Such verbs also have higher a case density value compared to the average value for verbs in the entire dataset. Almost all these verbs are perfective verbs which are transitive in nature. In Hindi, it is well known that the ergative marker *ne* indicates the presence of an upcoming transitive verb with perfective aspect (Choudhary et al., 2009; Husain et al., 2014). Vice-versa, we observe verbs having lower-than-average values for both average dependency length and case density. The verbs in this set are mostly auxiliary verbs like *hai* and *tha*. Thus root verbs with longer dependencies are associated with dependents marked by more case markers and conversely, verbs involved in shorter dependencies are linked to fewer case-marked heads.

In a recent work, Ranjan et al. (2022b) conjectured that the presence of semantically similar noun phrases and case markers are a strong factor potentially overriding the pressure for dependency length minimization in determining Hindi word-ordering choices. They further observed that the dependency length was an effective predictor of human choices only at very high dependency length values, consistent with prior work in the literature denoting interference effects in long-distance dependency resolution (Van Dyke and McElree, 2006; Van Dyke, 2007; Lewis, 1996). To substantiate these insights further, we fitted separate regression models containing all the predictors on four different bins of the dependency length; we report the results in Table 2. Our results at high dependency length suggest that long-distance dependency resolution is indeed driven by interference effects. However, case-marker effects are significant in all bins except

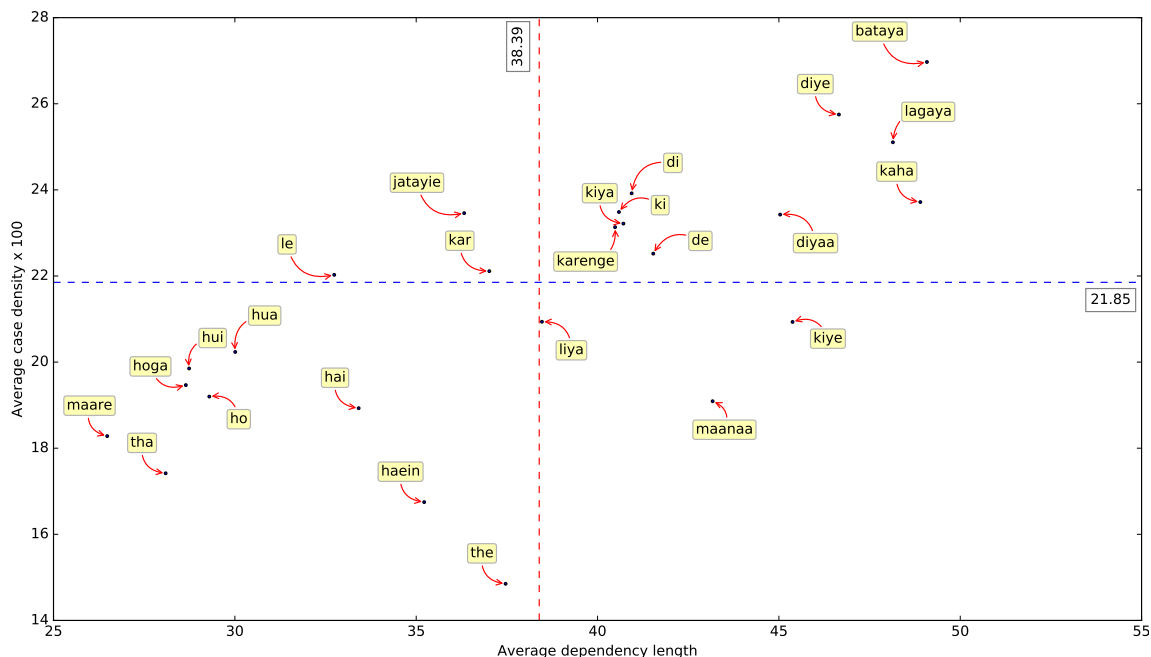


Figure 3: Average case density and dependency length for the 25 most-frequent HUTB verbs (average dependency length and case density values for the entire dataset depicted as dotted lines parallel to X and Y axes respectively)

the final one, a finding which requires more exploration factoring in the possibility of the interplay between case-based facilitation (Logačev and Vasishth, 2012) and cosine similarity. In other words, interference (whether proactive or retroactive) on account of a greater number of similar intervening items might be the working mechanism behind the processing difficulty postulated for longer dependency distances. This finding is also consistent with prior work in the literature, which argues that decay, the underlying cognitive construct behind locality, does not have robust empirical evidence supporting it (Engelmann et al., 2019; Oberauer and Lewandowsky, 2013, 2014; Stone et al., 2020; Berman et al., 2009, cf. Hardt et al., 2013).

4.2 Construction Analysis

In this section, we examined two Hindi syntactic constructions studied in the sentence processing literature, *viz.*, object-fronted (non-canonical) word orders, *i.e.*, direct (DO) and indirect object (IO) fronting (Vasishth, 2004), and conjunct verb constructions (Husain et al., 2014).

4.2.1 Non-canonical word order

We analyzed HUTB sentences that displayed Object-Subject-Verb (OSV) order, as illustrated in Example 1. The fronted objects could be di-

rect or indirect. Vasishth (2004) showed that dependency length effectively predicts the processing difficulties associated with OSV orders. Husain et al. (2014) demonstrated that sentences with conjunct verbs exhibit anti-locality effects. Table 3 displays the regression results for DO-/IO-fronted subsets and conjunct-verb constructions. For both DO- and IO-fronted sentences, our results revealed that the similarity measure is the only feature that significantly predicts the dependency length, while other effects are non-significant. The regression coefficient is also in the expected direction. Vasishth (2004) in his investigation of OSV order in Hindi reported that unlike IO-fronted sentences, the DO-fronted sentences still remained difficult to comprehend when provided with appropriate discourse context as compared to their canonical counterparts. He attributed the difficulty to greater dependency length, and thereby greater memory load, associated with DO-fronted sentences.

Example 2 illustrates a DO-fronted sentence from our dataset where all preverbal heads (*gundon* (*henchmen*), *hathiyaar* (*weapons*), *police*, *kshetra* (*area*), *giraftaar* (*arrest*)) directly linked to the main verb (*kiya*) are highlighted. This non-canonical sentence exhibits a greater dependency length (1.91) than average (1.83), indicating that

Predictor	dl <= 1.36 (#495)	1.36 <dl <= 1.80 (#555)	1.80 <dl <= 2.20 (#448)	dl >2.20 (#498)
Intercept	1.13	1.59	1.99	2.70
case density	0.03	0.01	0.02	NS
similarity	0.06	NS	0.01	0.07
IS Score	NS	-0.01	NS	-0.04
PCFG x 3.gram surprisal	-0.02	NS	NS	NS

Table 2: Four different regression models predicting average dependency length in binned data sets with the no. of data points in each indicated in column headers; column values represent regression coefficient of different predictors in the regression model; **dl = Average dependency length**; Bin-wise number of data points in parentheses; trigram and PCFG surprisal, and same case bigram features not shown as they are not significant (NS) in the models; Avg dl (Min, 1st Quartile, Mean, 3rd Quartile, Max) = 0.37, 1.36, 1.83, 2.20, 6.20

Predictor	$\hat{\beta}$	$\hat{\sigma}$	t
DO-FRONTED SUBSET			
Intercept	1.66	0.053	31.32
similarity	0.27	0.052	5.11
IO-FRONTED SUBSET			
Intercept	1.78	0.065	27.62
similarity	0.22	0.054	4.03
CONJUNCT-VERB SUBSET			
Intercept	1.93	0.021	92.24
PCFG surprisal	0.08	0.027	3.32
case density	0.09	0.022	4.13
similarity	0.18	0.019	9.34

Table 3: Linear regression model predicting average dependency length on DO-fronted (133), IO-fronted (101), and conjunct-verb (1158) data sets; significant predictors denoted in bold; non-significant predictors not shown here but see Appendix E for full details)

OSV sentences generally impose a higher memory load. Additionally, this sentence also has a higher semantic similarity (0.18) than the average (0.08) due to confusability among the four aforementioned preverbal head nouns (two animate and two inanimate nouns) when retrieved at the main verb. Therefore, these results suggest that the observed difficulty, as captured by dependency length in these OSV constructions, can be effectively explained by examining the semantic similarity among the preverbal heads within a sentence.

- (2) [kukhyaat sargana chhota rajan giroh-ke
Infamous gangster chhota raja gang-GEN
chaar **gundon**-ko]_{DO} **hathiyaar** sahit
four goons-ACC weapons along with
[**police**-ne]_S sehar-ke uttari paschami **kshetra**-se
police-ERG city-GEN north-western area-LOC
[**giraftaar**]_{POF} kiya
arrest do.PST.PFV.SG.M

The police arrested four henchmen from the no-

torious gangster Chhota Rajan gang, along with weapons, from the north-western area of the city.

4.2.2 Conjunct verbs

We focused on sentences in the corpus that contained noun-verb complex predicates, commonly referred to as conjunct verbs (Kachru, 1982; Butt, 1995; Mohanan, 1994). A conjunct verb consists of a complex predicate composed of a noun and a subsequent verb; these are annotated with the POF dependency relation in the HUTB corpus (See Example 5 in Appendix F).

For conjunct verb constructions (bottom block in Table 3), our analysis revealed that semantic similarity, case density, and PCFG surprisal emerged as significant predictors of dependency length. Notably, these predictors displayed positive regression coefficients, affirming the validity of our proposed hypothesis. In a self-paced reading study, Husain et al. (2014) found no significant reading time differences at the final verb in non-compositional sequences (e.g., *khyaal rakhna*) when the noun-verb distance increased with intervening adverbials. They observed locality effects only in simple predicates where the final verb was not predictable from its noun counterpart (e.g., *guitar rakhna*). Table 8 in Appendix G depicts construction-wise average feature values. In comparison to sentences with non-canonical word orders (11.72%), the conjunct verbs constructions (58.02%) are very frequent in our dataset and have higher average dependency length, number of constituents, similarity, and case density. Thus the differential impact of various features across the three constructions can be explained by the variation in these basic properties.

Thus, these construction-specific results further corroborate the view that the underlying reason behind locality effects may not be decay but rather more general memory retrieval and interference

effects as captured by semantic similarity.

5 Discussion

Our results show that our proposed interference measures, viz. semantic similarity and case density, model locality effects (as captured by dependency length) in Hindi. Their effects remain consistent at high dependency length, suggesting that dependency locality may be driven not just by decay of information but also by proactive and retroactive interference. Our findings also highlight that long dependencies involve a greater proportion of case markers. This reinforces the idea that within a natural corpus, the processing load on account of longer dependencies is due to increased memory load caused by interference and predictability effects arising from case markers. Additionally, we found that sentences with longer dependency lengths consistently exhibited high PCFG syntactic surprisal but low lexical trigram surprisal, with a notable interaction between the two. This hints at the possibility that syntactically complex sentences (as denoted by longer dependencies or greater syntactic surprisal) perhaps feature more probable word sequences, potentially mitigating the memory load. Finally, we noted that the interference due to same-case bigrams (*i.e.*, adjacent NPs marked with the same case marker) is insignificant in predicting dependency length, and further analyses confirmed that their effects were already accounted for by our semantic similarity measure.

For non-canonical OSV orders, we found that semantic similarity was the only significant positive predictor of dependency length. In contrast, for the sentences with conjunct verbs, in addition to PCFG surprisal, both semantic similarity and case density were significant predictors of dependency length, thereby validating our initial hypothesis across both constructions. These findings provide further insights for retrieval interference as an explanatory mechanism underlying locality effects that have been observed across various constructions in Hindi (Vasishth, 2004; Vasishth and Lewis, 2006; Husain et al., 2014; Ranjan et al., 2022b; Ranjan and von der Malsburg, 2023, 2024). Our results corroborate previous conjectures suggesting that temporal decay alone may not be the only explanation for the observed locality effects (Berman et al., 2009; Oberauer and Lewandowsky, 2013, 2014; Engelmann et al., 2019; Stone et al., 2020; Ranjan et al., 2022b, cf. Hardt et al., 2013).

More recently, Ranjan and van Schijndel (2024) in their extensive study of non-canonical word orders in a corpus of naturally occurring Hindi text demonstrated that discourse expectations captured by surprisal estimates from neural language models fine-tuned over preceding sentential context primarily govern the production of Hindi sentences. Notably, they report that discourse-enhanced surprisal entirely subsumes the impact of dependency length minimization effects in predicting Hindi OSV orders. Future work needs to investigate how interference, locality and surprisal jointly shape natural languages and human behaviour.

Our results provide an empirical basis for Vasishth (2011)'s theoretical proposal, where it was argued that locality and interference could be different manifestations of the same phenomenon. Vasishth contends that dependency locality instantiates the concept of decay in the form of dependency distance by counting the number of intervening discourse referents. In contrast, interference has no notion of memory limitation (storage) and only exhibits its effect through syntactic and semantic integration during retrieval processes, which get affected by the nature, quality, and specific content of information stored in the memory. Our semantic similarity measure (as quantified by cosine similarity among preverbal heads) significantly predicts the dependency length in Hindi, possibly indicating that interference effects may subsume the predictions of dependency locality. Therefore, we propose that interference effects also need to be factored in while developing a comprehensive theory of sentence processing.

As a part of future work, we plan to investigate the role of interference in presence of various other factors such as surprisal, locality, and discourse considerations. We also intend to tease apart the distance *vs.* interference effects by studying the nature of intervening material between the head and dependent units in a more controlled setup. Finally, future work should explore the impact of these factors on other languages to make cross-linguistic generalizations, as well as on language production using spoken datasets.

In sum, our results suggest a significant association between locality and interference effects, perhaps indicating that locality might be a surface phenomenon whose internal workings are driven by interference during memory retrieval.

Acknowledgements

We thank Marten van Schijndel, Shravan Vasishth, and the audiences of the Sentence Processing Colloquium at University of Potsdam and Cornell's C.Psyd group for their insightful comments on this work. We would also like to express our gratitude to the anonymous reviewers of CogSci-2022, HSP-2023, and SCiL-2024 for their helpful suggestions and feedback. Finally, the last two authors acknowledge the extramural funding provided by the Department of Science and Technology of India through the Cognitive Science Research Initiative (project no. DST/CSRI/2018/263).

References

- Rama Kant Agnihotri. 2007. *Hindi: An Essential Grammar*. Essential Grammars. Routledge.
- Arpit Agrawal, Sumeet Agarwal, and Samar Husain. 2017. [Role of expectation and working memory constraints in Hindi comprehension: An eyetracking corpus analysis](#). *Journal of Eye Movement Research*, 10(2).
- John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An Integrated Theory of the Mind. *Psychology Review*, 111(4):1036–1060.
- John R. Anderson and Rebecca Paulson. 1977. [Representation and retention of verbatim information](#). *Journal of Verbal Learning and Verbal Behavior*, 16(4):439 – 451.
- Serine Avetisyan, Sol Lago, and Shravan Vasishth. 2020. [Does case marking affect agreement attraction in comprehension?](#) *Journal of Memory and Language*, 112:104087.
- AD Baddeley and G Hitch. 1977. Recency re-examined. s. dornic (ed.) *attention and performance* (vol. 6, pp. 647-667).
- Paul Baker, Andrew Hardie, Tony McEnery, Hamish Cunningham, and Robert Gaizauskas. 2002. Emille: a 67-million word corpus of indic languages: data collection, mark-up and harmonization. In *Proceedings of LREC 2002*, pages 819–827. Lancaster University.
- Marc G Berman, John Jonides, and Richard L Lewis. 2009. In search of decay in verbal short-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(2):317.
- Kushagra Bhatia, Divyanshu Aggarwal, and Ashwini Vaidya. 2021. [Fine-tuning distributional semantic models for closely-related languages](#). In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 60–66, Kiyv, Ukraine. Association for Computational Linguistics.
- Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. [A multi-representational and multi-layered treebank for Hindi/urdu](#). In *Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP '09*, pages 186–189, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Marisa Ferrara Boston, John Hale, Reinhold Kliegl, Umesh Patil, and Shravan Vasishth. 2008. [Parsing costs as predictors of reading difficulty: An evaluation using the potsdam sentence corpus](#). *Journal of Eye Movement Research*, 2(1).
- Miriam Butt. 1995. *The structure of complex predicates in Urdu*. Center for the Study of Language (CSLI).
- Miriam Butt and Tracy Holloway King. 1996. Structural topic and focus without movement. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the First LFG Conference*. CSLI Publications, Stanford.
- Wallace Chafe. 1976. Givenness, contrastiveness, definiteness, subjects, topics, and point of view. *Subject and topic*.
- Kamal Kumar Choudhary, Matthias Schlesewsky, Dietmar Roehm, and Ina D. Bornkessel-Schlesewsky. 2009. [The N400 as a correlate of interpretively relevant linguistic rules: Evidence from Hindi](#). *Neuropsychologia*, 47(13):3012–3022.
- H. H. Clark and S. E. Haviland. 1977. Comprehension and the Given-New Contract. In R. O. Freedle, editor, *Discourse Production and Comprehension*, pages 1–40. Ablex Publishing, Hillsdale, N. J.
- Samvit Dammalapati, Rajakrishnan Rajkumar, Sidharth Ranjan, and Sumeet Agarwal. 2021. [Effects of duration, locality, and surprisal in speech disfluency prediction in english spontaneous speech](#). In *Proceedings of the Society for Computation in Linguistics*, volume 4, page 10.
- Simon De Deyne, Amy Perfors, and Daniel J Navarro. 2016. [Predicting human similarity judgments with distributional models: The value of word associations](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1861–1870.
- Vera Demberg and Frank Keller. 2008. [Data from eye-tracking corpora as evidence for theories of syntactic processing complexity](#). *Cognition*, 109(2):193–210.
- Herman Ebbinghaus. 1885. *Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie*. Duncker & Humblot.
- Hermann Ebbinghaus. 2013. [Memory: A contribution to experimental psychology](#). *Annals of neurosciences*, 20(4):155.
- Felix Engelmann, Lena A. Jager, and Shravan Vasishth. 2019. [The effect of prominence and cue association on retrieval processes: A computational account](#). *Cognitive Science*, 43(12):e12800.

- Stefan L Frank. 2017. [Word embedding distance does not predict word reading time](#). In *Proceedings of the 39th Annual Conference of the Cognitive Science Society (CogSci)*, pages 385–390. Cognitive Science Society, Austin.
- Richard Futrell. 2019. [Information-theoretic locality properties of natural language](#). In *Proceedings of the First Workshop on Quantitative Syntax (Quasy, SyntaxFest 2019)*, pages 2–15, Paris, France. Association for Computational Linguistics.
- Richard Futrell, Roger P Levy, and Edward Gibson. 2020. Dependency locality as an explanatory principle for word order. *Language*, 96(2):371–412.
- Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. [Large-scale evidence of dependency length minimization in 37 languages](#). *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.
- Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68:1–76.
- Edward Gibson. 2000. [Dependency locality theory: A distance-based theory of linguistic complexity](#). In Alec Marantz, Yasushi Miyashita, and Wayne O’Neil, editors, *Image, Language, brain: Papers from the First Mind Articulation Project Symposium*. MIT Press, Cambridge, MA.
- Peter C. Gordon, Randall Hendrick, and Marcus Johnson. 2001. [Memory interference during language processing](#). *Journal of Experimental Psychology: Learning Memory and Cognition*, 27(6):1411–1423.
- Peter C. Gordon, Randall Hendrick, Marcus Johnson, and Yoonhyoung Lee. 2006. [Similarity-based interference during language comprehension: Evidence from eye tracking during reading](#). *Journal of Experimental Psychology: Learning Memory and Cognition*, 32(6):1304–1321.
- Peter C. Gordon, Randall Hendrick, and William H. Levine. 2002. [Memory-load interference in syntactic processing](#). *Psychological Science*, 13(5):425–430. PMID: 12219808.
- Édouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomáš Mikolov. 2018. [Learning word vectors for 157 languages](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Alvin Grissom II, Naho Orita, and Jordan Boyd-Graber. 2016. [Incremental prediction of sentence-final verbs: Humans versus machines](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 95–104. Association for Computational Linguistics.
- John Hale. 2001. [A probabilistic Earley parser as a psycholinguistic model](#). In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies, NAACL ’01*, pages 1–8, Pittsburgh, Pennsylvania. Association for Computational Linguistics.
- Oliver Hardt, Karim Nader, and Lynn Nadel. 2013. Decay happens: the role of active forgetting in memory. *Trends in cognitive sciences*, 17(3):111–120.
- Samar Husain, Shrahan Vasishth, and Narayanan Srinivasan. 2014. [Strong expectations cancel locality effects: Evidence from Hindi](#). *PLOS ONE*, 9(7):1–14.
- Samar Husain, Shrahan Vasishth, and Narayanan Srinivasan. 2015. [Integration and prediction difficulty in Hindi sentence comprehension: Evidence from an eye-tracking corpus](#). *Journal of Eye Movement Research*, 8(2).
- T. Florian Jaeger and Elizabeth Norcliffe. 2009. [The cross-linguistic study of sentence production: State of the art and a call for action](#). *Language and Linguistic Compass*, 3(4):866–887.
- Lena A Jäger, Felix Engelmann, and Shrahan Vasishth. 2015. Retrieval interference in reflexive processing: Experimental evidence from mandarin, and computational modeling. *Frontiers in psychology*, 6:617.
- Lena A. Jager, Felix Engelmann, and Shrahan Vasishth. 2017. [Similarity-based interference in sentence comprehension: Literature review and bayesian meta-analysis](#). *Journal of Memory and Language*, 94:316 – 339.
- Y. Kachru. 2006. *Hindi*. London Oriental and African language library. John Benjamins Publishing Company.
- Yamuna Kachru. 1982. [Conjunct verbs in hindi-urdu and persian](#). *South Asian Review*, 6(3):117–126.
- Elsi Kaiser and John C Trueswell. 2004. The role of discourse context in the processing of a flexible word-order language. *Cognition*, 94(2):113–147.
- Sun-Hee Lee, Mineharu Nakayama, and Richard L. Lewis. 2005. Difficulty of processing Japanese and Korean center-embedding constructions. In M. Minami, H. Kobayashi, M. Nakayama, and H. Sirai, editors, *Studies in Language Science*, volume Volume 4, pages 99–118. Kurosio Publishers, Tokyo, Tokyo.
- Roger Levy. 2008. [Expectation-based syntactic comprehension](#). *Cognition*, 106(3):1126 – 1177.
- Roger Levy, Evelina Fedorenko, and Edward Gibson. 2013. [The syntactic complexity of russian relative clauses](#). *Journal of Memory and Language*, 69(4):461 – 495.
- R. L. Lewis and S. Vasishth. 2005. [An activation-based model of sentence processing as skilled memory retrieval](#). *Cognitive Science*, 29:1–45.

- Richard L Lewis. 1996. Interference in short-term memory: The magical number two (or three) in sentence processing. *Journal of psycholinguistic research*, 25(1):93–115.
- Richard L. Lewis. 1998. Interference in working memory: Retroactive and proactive interference in parsing. CUNY sentence processing conference.
- Richard L. Lewis and Mineharu Nakayama. 2001. Syntactic and positional similarity effects in the processing of Japanese embeddings. In *Sentence Processing in East Asian Languages*, pages 85–113, Stanford, CA. CSLI.
- Haitao Liu. 2008. [Dependency distance as a metric of language comprehension difficulty](#). *Journal of Cognitive Science*, 9(2):159–191.
- Pavel Logačev and Shravan Vasishth. 2012. [Case matching and conflicting bindings interference](#). In Monique Lamers and Peter de Swart, editors, *Case, Word Order and Prominence: Interacting Cues in Language Production and Comprehension*, pages 187–216. Springer Netherlands, Dordrecht.
- Paola Merlo and Francesco Ackermann. 2018. [Vectorial semantic spaces do not encode human judgments of intervention similarity](#). In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 392–401.
- Tara Mohanan. 1994. *Argument structure in Hindi*. Center for the Study of Language (CSLI).
- Elisabeth Norcliffe, Alice C Harris, and T Florian Jaeger. 2015. Cross-linguistic psycholinguistics and its critical role in theory development: Early beginnings and recent advances.
- Klaus Oberauer and Stephan Lewandowsky. 2013. [Evidence against decay in verbal working memory](#). *Journal of Experimental Psychology: General*, 142(2):380–411.
- Klaus Oberauer and Stephan Lewandowsky. 2014. Further evidence against decay in working memory. *Journal of Memory and Language*, 73:15–30.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. [Learning accurate, compact, and interpretable tree annotation](#). In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 433–440, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Sidharth Ranjan, Sumeet Agarwal, and Rajakrishnan Rajkumar. 2019. [Surprisal and interference effects of case markers in Hindi word order](#). In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 30–42, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sidharth Ranjan, Rajakrishnan Rajkumar, and Sumeet Agarwal. 2022a. [Linguistic Complexity and Planning Effects on Word Duration in Hindi Read Aloud Speech](#). In *Proceedings of the Society for Computation in Linguistics (SCiL)*, 5:11.
- Sidharth Ranjan, Rajakrishnan Rajkumar, and Sumeet Agarwal. 2022b. [Locality and expectation effects in Hindi preverbal constituent ordering](#). *Cognition*, 223:104959.
- Sidharth Ranjan and Marten van Schijndel. 2024. Does dependency locality predict non-canonical word order in Hindi? In *Proceedings of the 46th Annual Meeting of the Cognitive Science Society*, Rotterdam, Netherlands. Cognitive Science Society, Cognitive Science Society.
- Sidharth Ranjan and Titus von der Malsburg. 2023. [A bounded rationality account of dependency length minimization in Hindi](#). In *Proceedings of the 45th Annual Meeting of the Cognitive Science Society*, Sydney, Australia. Cognitive Science Society, Cognitive Science Society.
- Sidharth Ranjan and Titus von der Malsburg. 2024. [Work smarter...not harder: Efficient minimization of dependency length in SOV languages](#). In *Proceedings of the 46th Annual Meeting of the Cognitive Science Society*, Rotterdam, Netherlands. Cognitive Science Society, Cognitive Science Society.
- Brian Roark, Asaf Bachrach, Carlos Cardenas, and Christophe Pallier. 2009. [Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing](#). In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 324–333, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Henry L Roediger III and Magdalena Abel. 2022. The double-edged sword of memory retrieval. *Nature Reviews Psychology*, 1(12):708–720.
- Lavinia Salicchi, Alessandro Lenci, and Emmanuele Chersoni. 2021. [Looking for a role for word embeddings in eye-tracking features prediction: Does semantic similarity help?](#) In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 87–92, Groningen, The Netherlands (online). Association for Computational Linguistics.
- Gerard Salton. 1972. [A new comparison between conventional indexing \(medlars\) and automatic text processing \(smart\)](#). *Journal of the American Society for Information Science*, 23(2):75–84.
- C. E. Shannon. 1948. A mathematical theory of communication. *Bell system technical journal*, 27.
- Kartik Sharma, Richard Futrell, and Samar Husain. 2020. What determines the order of verbal dependents in hindi? effects of efficiency in comprehension and production. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–10.

- Garrett Smith and Shravan Vasishth. 2020. [A principled approach to feature selection in models of sentence processing](#). *Cognitive science*, 44(12):e12918.
- Adrian Staub. 2010. Eye movements and processing difficulty in object relative clauses. *Cognition*, 116:71–86.
- Andreas Stolcke. 2002. SRILM — An extensible language modeling toolkit. In *Proc. ICSLP-02*.
- Kate Stone, Titus von der Malsburg, and Shravan Vasishth. 2020. The effect of decay and lexical uncertainty on processing long-distance dependencies in reading. *PeerJ*, 8:e10438.
- David Temperley. 2007. [Minimization of dependency length in written English](#). *Cognition*, 105(2):300–333.
- Harry Tily. 2010. *The Role of Processing Complexity in Word Order Variation and Change*. Ph.D. thesis, Stanford University. Unpublished thesis.
- Matthew J Traxler, Robin K Morris, and Rachel E Seely. 2002. Processing subject and object relative clauses: Evidence from eye movements. *Journal of memory and language*, 47(1):69–90.
- Julie A. Van Dyke. 2007. Interference effects from grammatically unavailable constituents during sentence processing. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, 33 2:407–30.
- Julie A. Van Dyke and Clinton L. Johns. 2012. [Memory interference as a determinant of language comprehension](#). *Language and Linguistics Compass*, 6(4):193–211.
- Julie A. Van Dyke and R. L. Lewis. 2003. Distinguishing effects of structure and decay on attachment and repair: a retrieval interference theory of recovery from misanalyzed ambiguities. *Journal of Memory and Language*, 49:285–413.
- Julie A. Van Dyke and Brian McElree. 2006. [Retrieval interference in sentence comprehension](#). *Journal of Memory and Language*, 55(2):157 – 166.
- S. Vasishth. 2003. *Working Memory in Sentence Comprehension: Processing Hindi Center Embeddings*. Outstanding Dissertations in Linguistics. Taylor & Francis.
- S. Vasishth. 2004. [Discourse context and word order preferences in Hindi](#). *Yearbook of South Asian Languages*, pages 113–127.
- Shravan Vasishth. 2011. Integration and prediction in head-final structures. In Yuki Hirose and Jerome L. Packard, editors, *Processing and Producing Head-final Structures*, pages 349–367. Springer Netherlands, Dordrecht.
- Shravan Vasishth and Heiner Drenhaus. 2011. Locality in German. *Dialogue & Discourse*, 2(1):59–82.
- Shravan Vasishth and Richard L. Lewis. 2006. [Argument-head distance and processing complexity: Explaining both locality and antilocality effects](#). *Language*, 82(4):767–794.
- Himanshu Yadav, Ashwini Vaidya, and Samar Husain. 2017. Keeping it simple: Generating phrase structure trees from a Hindi dependency treebank. In *TLT*.
- Victor H Yngve. 1960. A model and an hypothesis for language structure. *Proceedings of the American philosophical society*, 104(5):444–466.

Appendix

A Dependency length calculation

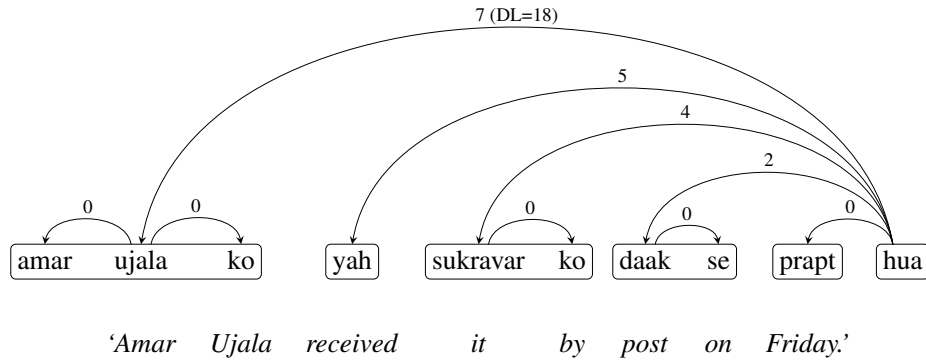


Figure 4: Calculation of dependency length in a dependency tree; Total dependency length (DL) of the structure indicated above the top arc; Word’s dependency length is mentioned above each dependency arc

B Information Status Annotation

(3) Preceding context sentence

amar ujala-ki bhumika nispaksh rehti hai
 Amar Ujala-GEN role unbiased remain be.PRS.SG

Amar Ujala’s role remains unbiased.

(4) Target Sentence

- a. [amar ujala-ko]_O [yah]_S sukravar-ko daak-se prapt hua [Given-Given = 0]
 Amar Ujala-ACC it friday-on post-INST receive be.PST.SG
 Amar Ujala received **it** by post on *Friday*.

In the above target example, the object phrase shares a content word “amar ujala” with the preceding context sentence. Therefore, the object phrase is assigned a GIVEN tag. Additionally, the subject phrase “yah” in the target sentence is a pronoun, so it is also assigned a GIVEN tag. As a result, the target sentence, overall, belongs to GIVEN-GIVEN ordering.

C Hindi Case Markers

Marker	Case (Gloss)	Grammatical Function
ϕ	nominative (NOM)	subject/object
ne	ergative (ERG)	subject
ko	accusative (ACC)	object
	dative (DAT)	subject/indirect object
se	instrumental (INS)	subject/oblique/adjunct
$ka/ki/ke$	genitive (GEN)	subject (infinitives) specifier
$m\tilde{e}/par/tak$	locative (LOC)	oblique/adjunct

Table 4: Hindi case markers (Butt and King, 1996)

D Pearson Correlation Analysis

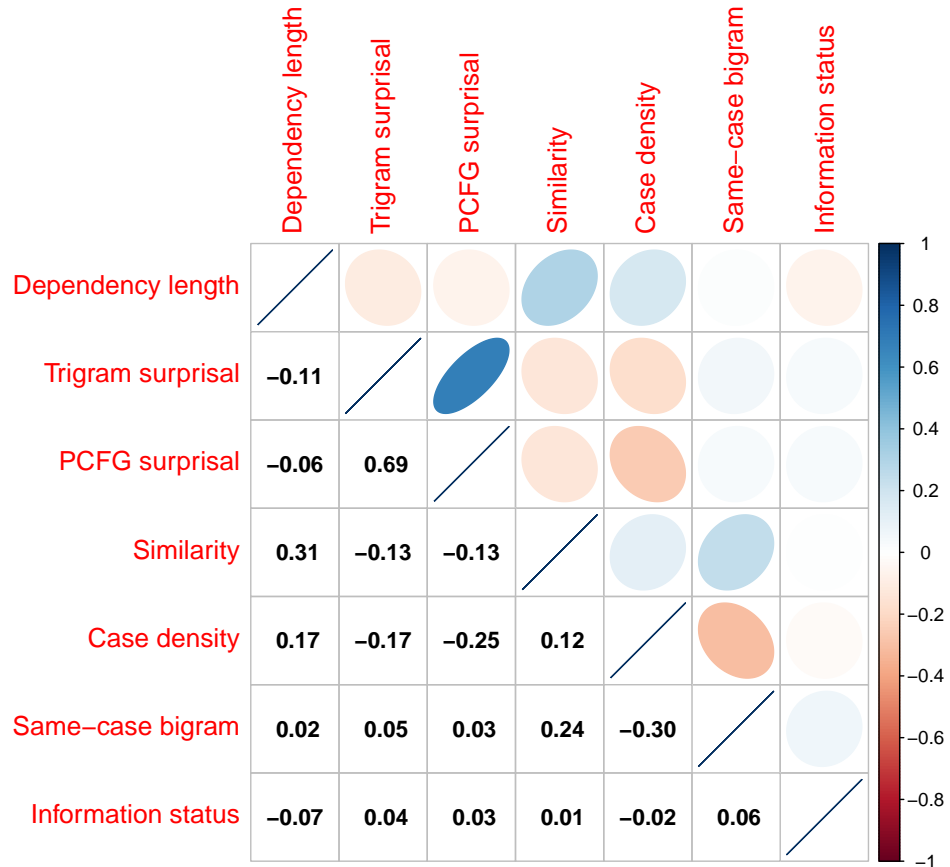


Figure 5: Pearson's correlation coefficient between various cognitive measures

E Construction Analysis

Predictor	$\hat{\beta}$	$\hat{\sigma}$	t
Intercept	1.66	0.053	31.32
IS Score	0.02	0.044	0.35
PCFG surprisal	0.06	0.066	0.88
3.gram surprisal	-0.01	0.064	-0.15
same case bigram	-0.01	0.052	-0.27
case density	0.08	0.052	1.57
similarity	0.27	0.052	5.11
3.gram x PCFG surp	-0.05	0.034	-1.31

Table 5: Linear regression model predicting average dependency length on DO-fronted dataset (133); significant predictors denoted in bold; other predictors not shown as they are not significant in the model

Predictor	$\hat{\beta}$	$\hat{\sigma}$	t
Intercept	1.78	0.065	27.62
IS Score	0.02	0.050	0.40
PCFG surprisal	0.04	0.081	0.48
3.gram surprisal	0.03	0.084	0.40
same case bigram	-0.01	0.060	-0.14
case density	-0.04	0.058	-0.65
similarity	0.22	0.054	4.03
3.gram x PCFG surp	-0.07	0.045	-1.59

Table 6: Linear regression model predicting average dependency length on IO-fronted dataset (101); significant predictors denoted in bold; other predictors not shown as they are not significant in the model

Predictor	$\hat{\beta}$	$\hat{\sigma}$	t
Intercept	1.93	0.021	92.24
IS Score	-0.02	0.018	-1.54
PCFG surprisal	0.08	0.027	3.32
3.gram surprisal	-0.05	0.027	-1.64
same case bigram	-0.03	0.019	-1.57
case density	0.09	0.022	4.13
similarity	0.18	0.019	9.34
3.gram x PCFG surp	-0.03	0.02	-1.72

Table 7: Linear regression model predicting average dependency length on conjunct-verb dataset (1158); significant predictors denoted in bold; other predictors not shown as they are not significant in the model

F Conjunct Verb Construction

In Hindi conjunct verbs, a highly predictable verb follows a nominal element, resulting in a non-compositional meaning such as *khyaal rakhna* (‘care keep/put’; ‘to take care of’) as opposed to *guitar rakhna* (‘guitar keep/put’; ‘to put down or keep a guitar’). The following example illustrates Hindi conjunct verbs:

- (5) baasu chatterjee-ne apne parivaar-ka [khyaal]_{POF} rakha
baasu chatterjee-ERG his own family-GEN care keep.PST.PFV
Basu Chatterjee took care of his family.

G Dataset distribution

Construction(#cases)	DL	Similarity	Case density	Same-Same Sequence	Trigram surprisal	PCFG surprisal	Sentence length	#Preverbal constituents
Conjunct verbs (1158)	46.40	0.42	0.21	0.49	49.26	138.97	22.42	4.28
IO-fronted orders (101)	38.73	0.35	0.21	0.29	45.71	126.19	20.33	3.66
DO-fronted orders (133)	29.56	0.31	0.19	0.46	41.53	112.54	17.08	3.69
Full data (1996)	40.04	0.38	0.21	0.44	45.03	125.93	20.03	4.04

Table 8: Construction-specific statistics (mean values)

Online Learning of ITSL Grammars

Jacob K. Johnson

Kahlert School of Computing
University of Utah

`jacob@nnnNNnnn.info`

Aniello De Santo

Dept. of Linguistics
University of Utah

`aniello.desanto@utah.edu`

Abstract

This paper presents the first incremental learning algorithm for input-sensitive TSL languages (ITSL). We leverage insights from De Santo and Aksënova (2021)'s ITSL batch-learner to generalize Lambert (2021)'s string extension learning approach to online learning of TSL. We discuss formal properties of the extension, and evaluate the effectiveness of both the original TSL learner and the new ITSL learner on a variety of phonotactic patterns.

1 Introduction

In the mathematical study of linguistic dependencies, the subregular (McNaughton and Papert, 1971; Heinz, 2011a,b; Chandlee and Heinz, 2016) class of Tier-based Strictly Local languages (TSL; Heinz et al., 2011) has gained prominence due to its ability to account for a variety of local and long-distance phonotactic phenomena. TSL as a formal class draws its linguistic inspiration from autosegmental phonology (Goldsmith, 1976), and it is characterized by two components: i) strictly local constraints on adjacent segments, and ii) a tier projection mechanism selecting string elements from a subset of the alphabet over which to enforce such constraints. Long-distance dependencies are thus thought of as local dependencies over strings where irrelevant segments (i.e. segments not part of the alphabet subset) are masked out.

From a typological perspective, the relativized adjacency at the core of the tier-based local constraints has made the TSL class fruitful in characterizing a vast amount of both local and unbounded phonotactic phenomena (McMullin, 2016, a.o.), and tier-locality has been proposed as a general mechanism to account for unbounded processes across linguistic domains (Aksënova et al., 2016; Vu et al., 2019; Graf, 2022a,b). Additionally, a variety of extensions of TSL have been proposed that take advantage of the relativized adjacency intuition while enriching the way elements of the tier are selected (Mayer and Major, 2018; Graf and Mayer, 2018; De Santo and Graf, 2019).

In particular, De Santo and Graf (2019) observe that by conditioning tier-membership not just on the identity of a symbol, but also on its local contexts (which symbols precede or follow it) it is possible to generalize TSL to a class of languages (*input-sensitive* TSL, or ITSL) capturing the interaction of local and non-local processes simultaneously.

From a learnability perspective, TSL has been shown to be efficiently learnable in the limit from positive data only (Gold, 1967), assuming a batch-learning set-up where all data are fed to the learner at once (Jardine and Heinz, 2016; Jardine and McMullin, 2017) and more recently also in an online learning setting (Lambert, 2021). As Lambert (2021) observes, online learning of subregular classes seems to be a fundamental step in exploiting mathematical insights to develop learning algorithms that are plausible from a human perspective — given that batch learning assumes simultaneous access to all prior input. Moving beyond TSL, De Santo and Aksënova (2021) propose an efficient batch learning algorithm for (multiple) ITSL grammars with tier-constraints bounded to $k=2$, extending the TSL learner of Jardine and Heinz (2016) and the multiple TSL learner of McMullin et al. (2019). In this work, we leverage the insights of De Santo and Aksënova (2021), and we show how a minor modification to the definition of symbol allows us to extend Lambert (2021)'s TSL learner to an online ITSL learner. We thus contribute the first online learning algorithm for ITSL, including an open source Python 3 implementation of both the new ITSL extension and the original online TSL learner. We also follow the lead of Aksënova (2020) and Johnson and De Santo (2023), and offer a preliminary evaluation of the performance of both algorithms on data representing a variety of phonotactic patterns.

We start with some formal preliminaries (Section 2) necessary to ground our modification of the work in Lambert (2021), and provide some background on TSL and ITSL in Section 3. Section 4 presents the core intuitions behind the existing online TSL

learner, and then our extension to an ITSL learner. Finally, we conduct a preliminary evaluation of both algorithms on natural and artificial datasets (Section 5), and conclude with a broader discussion of current results and future steps.

2 Notation and Terminology

We assume familiarity with set notation, specifically the union operator \cup , the element operator \in , the subset operator \subseteq , and the power set function $\mathcal{P}(\cdot)$. Sets are denoted as surrounded by curly braces $\{\}$, whereas angle brackets $\langle \rangle$ are used for ordered tuples.

Σ is used to denote some finite set of symbols, the alphabet. Σ^* is the set of all strings of finite length that can be formed using 0 or more instances of symbols from Σ . $\Sigma^k \subseteq \Sigma^*$ denotes the set of all strings that can be formed using exactly k instances of symbols from Σ . Likewise, $\Sigma^{\leq k} \subseteq \Sigma^*$ denotes the set of all strings that can be formed using k or fewer instances of symbols from Σ .

A language L is some subset of Σ^* . A grammar G can be thought of as a way to determine membership of a string in a stringset. If we denote the language associated with a grammar G as $L(G)$, G can be defined as a function to determine, for any string w , whether $w \in L(G)$.

In this paper, strings are denoted in monospace font, and ε denotes the empty or 0-length string. $|w|$ indicates the length of, or number of symbols in, a string w . The variable σ is commonly used to represent individual symbols, while the variables u, v, w, x, y are commonly used to represent strings. The concatenation of strings u and v , denoted uv , is the simple concatenation of the sequence of characters making up that string. That is, given $u = ab$ and $v = cd$, the concatenation $uv = abcd$. Concatenation is notated identically for individual symbols: given $\sigma_1 = e, \sigma_2 = f$, $uv\sigma_1\sigma_2 = abcdef$.

A string u is a substring of a string w iff $\exists x, y \in \Sigma^*$ such that $xuy = w$. Intuitively, this means that u is a substring of w if w contains u within it, without skipping or reusing symbols. A string $u = \sigma_1 \dots \sigma_{|u|}$ is a subsequence of a string w iff $\exists x_1, \dots, x_{|u|+1} \in \Sigma^*$ such that $\sigma_1 x_1 \dots x_{|u|} \sigma_{|u|} x_{|u|+1} = w$. Intuitively, this means that u is a subsequence of w if w contains u within it, without reusing symbols.

3 Background: TSL and ITSL

As mentioned, TSL (Heinz et al., 2011) formalizes the linguistic notion of a phonological tier (Goldsmith, 1976). We can think of a tier T as a subset (e.g.,

only sibilants) of the original alphabet available to a language. Then, given a string w , tier projection can be understood as forming a relativized locality domain by “masking out” all segments in w that do not belong to the tier alphabet, while preserving the ordering relations among segments in T . Long-distance dependencies (restrictions over segments that are non adjacent in the original string) can then be characterized as local dependencies within such relativized domain, and can thus be enforced by strictly local constraints of width k (i.e. k -grams). In terms of its fundamental components then, TSL is parameterized by the width (k) of the tier-constraints and by T — which defined the elements that are relevant to the dependencies. The interested reader is referred to Lambert and Rogers (2020) for a detailed characterization of this class in terms of model and automata theory, as well as to De Santo and Graf (2019) and Lambert and Rogers (2020) for a discussion of its closure properties.

While TSL has been shown to provide insightful characterizations for a variety of unbounded dependencies (Heinz et al., 2011; McMullin, 2016; Graf, 2017, a.o.), phonotactic studies cross-linguistically have revealed substantial limits to its expressivity tied to its projection mechanism — how tier-membership is evaluated (McMullin, 2016; Mayer and Major, 2018; Baek, 2017; Graf and Mayer, 2018; De Santo and Graf, 2019).

For example, in the Ineseño Chumash language of Southern California, a regressive sibilant harmony with unbounded locality ($[s]$ and $[ʃ]$ may not co-occur anywhere within the same word, cf. a) overrides a restriction against string-adjacent *st, *sn, *sl that results in a pattern of dissimilation (Applegate, 1972; McMullin, 2016). For instance, /sn/ surfaces as $[ʃn]$ (cf. b, c) unless there is an $[s]$ following in the string, in which case it surfaces as $[sn]$ (cf. d):

1) Unbounded sibilant harmony

a. /k-su-fojin/ kʃufojin “I darken it”

2) /s/ → [ʃ] when preceding (adjacent) [t, n, l]

b. /s-niʔ/ ʃniʔ “his neck”

c. /s-nanʔ/ ʃnanʔ “he goes”

3) Long-distance harmony overrides palatalization

d. /s-net-us/ snetus “he does it to him”

Figure 1 exemplifies why this overall pattern, involving an interaction of local and non-local constraints, is not TSL. Since $[sn]$ is sometimes observed in a string-adjacent context (as in d), it must

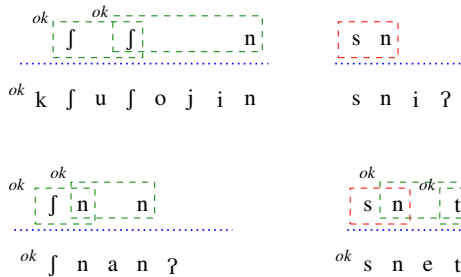


Figure 1: Example of a failed TSL analysis of Ineseño Chumash, adapted from De Santo and Graf (2019). Sibilants and $[t,n,l]$ are tier symbols.

be permitted as a 2-gram on a tier — even though it is only allowed when a segment such as $[s]$ follows them later in the string. But then, a TSL grammar would have no means of banning $*sn$ when there is no subsequent $[s]$ in the string. Vice-versa, if we ban $*sn$ on T , then the grammar will not be able to allow it when another $[s]$ follows on the tier. Additionally, we might point out that the difference between (b) and (d) could be resolved by extending the tier-grammar to consider 3-grams. However, in order to ban $*sn$, every occurrence of $[n]$ in the string must be projected on the tier (and in fact, to really capture the generalization, every occurrence of t and l too). Since the number of $[n,t,l]$ segments between two sibilants is potentially unbounded, no TSL grammar can generally account for this pattern, independently of the dimension of the tier k -grams.

In light of this, De Santo and Graf (2019) suggest to approach such limit by extending the locality window of the TSL projection. The m -Input-sensitive Tier-based Strictly k -Local (m -ITSL $_k$) class is thus defined by allowing the projection mechanism to consider the m -local context of a segment (i.e., its local surrounding environment) before projecting it on a tier.

See Figure 2, adapted from De Santo and Graf (2019), for a sketch of how this approach allows us to characterize the Ineseño Chumash pattern: by increasing the locality of the projection to 2 the grammar is allowed to project $[n]$ iff it is immediately preceded by a sibilant in the input string, and then use 3-local tier constraints to ban $\{*sn(\neg s), *fn_s\}$, in addition to the factors needed to enforce the usual sibilant harmony patterns. Thus, the possible unboundedness of $/n/$ is not a problem, since $/n/$ is now relevant for the projection only when adjacent to a sibilant.

More formally, ITSL is characterized by establishing tier-projection as an input strictly local function (Chandlee and Heinz, 2018) over m segments. This

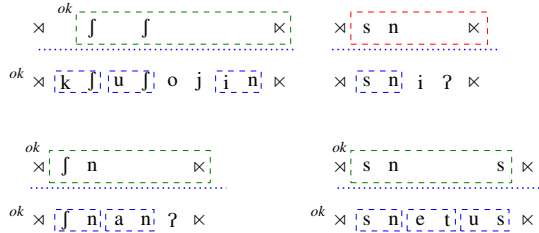


Figure 2: Example of an ITSL analysis of Ineseño Chumash, adapted from De Santo and Graf (2019). Assume a 2-local projection and 3-local tier constraints. $[n,t,l]$ are projected on the tier only when immediately preceded by $[s,f]$. \times and \times are word-boundaries.

modification takes full advantage of the original definition of tier-projection by Heinz et al. (2011), and extends TSL to a class that retains all its well-behaved subregular properties while significantly increasing its typological coverage.

From a learnability perspective, learning a TSL grammar from data alone implies being able to infer not only the relevant constraints, but also (or especially) the content of T . This is even more challenging for ITSL, since it relies on a more complex mechanism to establish tier membership. However, TSL $_k$ and m -ITSL $_k$ grammars have both been shown to be efficiently (polynomial in time and input) learnable in the limit from positive data in the sense of Gold (1967), even when the tier-alphabet is not known a priori (Jardine and Heinz, 2016; Jardine and McMullin, 2017; De Santo and Aksénova, 2021). Additionally, Lambert (2021) has recently proposed an algorithm for incremental learning of TSL. In what follows, we overview Lambert (2021)’s TSL learner and show how it can be extended to ITSL.

4 Online Learning Algorithms

Following Lambert (2021), we restrict this discussion to learning in the limit in the sense of Gold (1967). In this sense, given a set of strings $W \subseteq L(G)$, for some target grammar G , a learner function φ should output a learned grammar G' which is equivalent to G for sufficient data. This discussion is also restricted to incremental learning specifically. That is, the learner does not consider all the input at once, but instead evaluates a single item from the input (and a previously proposed grammar) at each step.

Lambert (2021) develops an incremental learner for TSL in the style of Heinz (2010). Heinz (2010) overviews several subregular classes of string languages sharing the common property that each string

in the language can be mapped to an element in the grammar, in terms of general String Extension Learners (McNaughton and Papert, 1971; Simon, 1975; Rogers and Pullum, 2011). The fundamental intuition centers around the notion of a *factor*, a connected substructure of a string. For strictly local languages (McNaughton and Papert, 1971), for example, a factor is an adjacent sequence of symbols. A strictly local language is the set of all strings containing only allowed (or not containing any forbidden) factors — and it is fully characterized by the set of all factors. Lambert (2021) exploits the fact that this property extends to TSL, where factors need to be defined over a subset of *salient* symbols in the alphabet (the tier), to provide a structural representation of TSL grammars that naturally lends itself to online learning efficiency in the learning paradigm of Gold (1967).

4.1 Learning TSL Online (Lambert, 2021)

Lambert (2021) identifies two core components of a TSL_k grammar that need to be identified by the learner, when T is not provided a priori: a) the set of underlying constraints (the strictly local factors of width k and b) the set of symbols that are *salient* to such constraints — the elements in T . In order to infer which elements belong to T then, Lambert (2021) relies two necessary and sufficient properties of any element *not* in T : free insertion and free deletion (Lambert and Rogers, 2020). Intuitively, if a segment is not salient to tier-constraints, that it is essentially invisible to the grammar: that is, there should be no way to restrict its distribution. Thus, such elements should be freely insertable and freely deletable in all strings without chance of affecting the well-formedness of such strings with respect to the grammar. Once the set of salient symbols has been defined, the learner can then infer strictly local constraints over the input filtered of irrelevant symbols.

Formalizing these observations, a TSL grammar is represented as the pair $\langle G_\ell, G_s \rangle$, where G_ℓ is the set of *attested factors* of width bounded above by $k+1$ and G_s is the set of *augmented subsequences* of length bounded above by k .

For a given k , the set of attested factors can be used to define salience, and it is bound to $k+1$ so to allow for an evaluation of both free insertability (so adding one symbol to k) and free deletability. For instance, the set of attested factors for a TSL_2 grammar for the string $cabacba$ is: $\{\varepsilon, a, b, c, ab, ac, ba, ca, cb, aba, acb, bac, cab, cba\}$.

As mentioned, once the set of salient symbols has

Subsequence	Intervener Sets
ε	$\{\{\}\}$
a	$\{\{\}\}$
b	$\{\{\}\}$
c	$\{\{\}\}$
aa	$\{\{b\}, \{b, c\}\}$
ab	$\{\{\}, \{c\}\}$
ac	$\{\{\}\}$
ba	$\{\{\}\}$
bb	$\{\{a, c\}\}$
bc	$\{\{a\}\}$
ca	$\{\{\}, \{b\}\}$
cb	$\{\{\}, \{a\}\}$
cc	$\{\{a, b\}\}$

Table 1: Augmented Subsequences extracted from $cabacba$ by a TSL learner with $k=2$, example adapted from Lambert (2021).

been detected, the next step is to infer the relevant k -local constraints. In batch learning, it would be possible to do a first pass over the input to infer tier-membership, and then a second pass over the same input with all non-salient symbols masked out in order to select constraints. However, in an online setting performing a second pass on the input would require to retaining every observed item, thus resulting in unbounded space requirements. Lambert (2021) gets around this obstacle by relying on the notion of *augmented subsequences*.

A subsequence is a factor over relativized adjacency: that is, a sequence of symbols that appear in order but not necessarily adjacent to each other. An augmented subsequence is a pair consisting of an attested subsequence, and a set of symbols that is attested to intervene among elements of such subsequence. Importantly, the same symbol cannot be both part of the subsequence and of the intervening set. To illustrate this concept, the set of attested augmented subsequences for a TSL_2 grammar for the string $abbacb$ is in Table 1.

For a given width k the space requirement to store all possible subsequences would still be exponential in the size of the alphabet and k . However, Lambert (2021) observes that storing all augmentations is in fact not necessary, due to subsumption relations between interveners. Consider for example the subsequence ab as attested in $cabacba$. Possible interveners for $cabacba$ are both $\{\}$ and $\{c\}$. But for $\{\}$ to be in the intervener set, it means that a and b can be immediately adjacent to each other: then, it does not matter how adjacency is relativized. That is, if $\{\}$ is an intervener then $\{c\}$ trivially also is, and we do

not need to maintain both. This observation generalizes to any subset/superset relation between intervener sets, so that the learner only has to maintain the smallest observed ones (partially-ordered by subset).

We can now overview the full procedure of the Online TSL learner. Initially, the learner assumes an empty grammar (represented as $\langle\{\},\{\}\rangle$). For each input string w from the language, the learner updates the grammar, making use of the following functions:

- $f : \Sigma^* \rightarrow \mathcal{P}(\Sigma^{\leq k+1})$ extracts all factors of w of width $\leq k+1$
- $x : \Sigma^* \rightarrow \mathcal{P}(\Sigma^{\leq k} \times \mathcal{P}(\Sigma))$ extracts the valid augmented subsequences of width $\leq k$
- $r : \mathcal{P}(\Sigma^{\leq k} \times \mathcal{P}(\Sigma)) \rightarrow \mathcal{P}(\Sigma^{\leq k} \times \mathcal{P}(\Sigma))$ removes all augmented subsequences that are already entailed by other augmented subsequences, that is, $r(S) \subseteq S$

The learning function $\varphi : \langle \mathcal{P}(\Sigma^{\leq k+1}) \times \mathcal{P}(\Sigma^{\leq k} \times \mathcal{P}(\Sigma)) \rangle \times \Sigma^* \rightarrow \langle \mathcal{P}(\Sigma^{\leq k+1}) \times \mathcal{P}(\Sigma^{\leq k} \times \mathcal{P}(\Sigma)) \rangle$ is as follows:

$$\varphi(\langle G_\ell, G_s \rangle, w) = \langle G_\ell \cup f(w), r(G_s \cup x(w)) \rangle$$

To recap, for each string w as input to the learner, G_ℓ is updated to $G_\ell \cup f(w)$ (that is, the factors of w of width $\leq k+1$ are added to G_ℓ), and G_s is updated to $r(G_s \cup x(w))$ (that is, the augmented subsequences of w of width $\leq k$ are added to G_s and any redundancy is removed). Table 2 shows an example of how the grammar is updated as the learner receives strings one by one.

4.2 Generalizing to Online ITSL Learning

To generalize the algorithm presented above to ITSL grammars, it is worth contrasting the formal definitions of the projection function for TSL and ITSL. As discussed, TSL languages have k -local constraints only apply to elements of a tier $T \subseteq \Sigma$. A projection function (also called erasing function) is thus defined as deleting (or masking) all symbols that are not in T .

Definition 1 (TSL Proj.; Heinz et al. (2011))

$$E_T(\sigma) := \begin{cases} \sigma & \text{if } \sigma \in T \\ \varepsilon & \text{otherwise} \end{cases}$$

In order to extend the notion of tier in TSL languages to consider local properties of the segments in the input string, De Santo and Graf (2019) follow Chandlee and Heinz (2018) and define an input-sensitive projection function in terms of local contexts (segments adjacent to a target symbol within a context window of width m).

Definition 2 (Contexts; De Santo and Graf (2019))

A m -context c over alphabet Σ is a triple σ, u, v such that $\sigma \in \Sigma$, $u, v \in \Sigma^*$ and $|u| + |v| \leq m$. A m -context set is a finite set of m -contexts.

Definition 3 (ISL Proj.; De Santo and Graf (2019))

Let C be a m -context set over Σ (where Σ is an arbitrary alphabet also containing edge-markers \times, \bowtie). Then the input strictly m -local (ISL- m) tier projection π_C maps every $s \in \Sigma^*$ to $\pi'_C(\times^{m-1}, s \times^{m-1})$, where $\pi'_C(u, \sigma v)$ is defined as follows, given $\sigma \in \Sigma \cup \varepsilon$ and $u, v \in \Sigma^*$:

$$\begin{aligned} \varepsilon & \quad \text{if } \sigma u v = \varepsilon, \\ \sigma \pi'_C(u, \sigma v) & \quad \text{if } \sigma, u, v \in C, \\ \pi'_C(u, \sigma v) & \quad \text{otherwise.} \end{aligned}$$

In essence, the notion of tier in ITSL is expressed by the set of contexts C , which is the set of tier segments augmented with locality conditions necessary for them to be salient to the tier constraints. Note also that an ISL-1 tier projection only determines projection of σ based on σ itself, showing that this projection function is really just an extension of what happens for TSL languages.

From an algorithmic perspective then, De Santo and Aksënova (2021) observe that having to evaluate salience of tier-segments based on m -local contexts (thus a segment plus its $m-1$ left or right context) can be understood as treating m -grams as unitary elements of the language. Thus, if we characterize every structure previously defined over individual segments over this more complex definition of symbol instead, we can directly lift the rest of the inference procedures for TSL. With this in mind, we can generalize the existing TSL online learning to ISTL in the same way De Santo and Aksënova (2021) generalized TSL batch learning.

For an m -ITSL learner, rather than considering w to be a string of $|w|$ symbols $\sigma_1, \dots, \sigma_{|w|} \in \Sigma$, we consider it a string of width- m overlapping substrings of w : $\sigma_{1\dots m}, \sigma_{2\dots m+1}, \dots, \sigma_{|w|-m+1\dots |w|} \in \Sigma^m$. We can then apply the TSL learning algorithm as sketched above, unchanged.

To illustrate these concepts, consider an ITSL grammar with $m = 2$ (the contexts) and $k = 2$ (the tier constraints): the string $cabacba$ is represented as $\langle ca, ab, ba, ac, cb, ba \rangle$. Thus the set of attested factors for $k = 2$ becomes: $\{\langle \rangle, \langle ab \rangle, \langle ac \rangle, \langle ba \rangle, \langle ca \rangle, \langle cb \rangle, \langle ab, ba \rangle, \langle ac, cb \rangle, \langle ba, ac \rangle, \langle ca, ab \rangle, \langle cb, ba \rangle, \langle ab, ba, ac \rangle, \langle ac, cb, ba \rangle, \langle ba, ac, cb \rangle, \langle ca, ab, ba \rangle\}$ (recall again that at this step we collect factors up to width $k+1$). Note that each unary symbol is now actually

w	G_ℓ	G_s
cabacba	$\{\varepsilon, a, b, c, ab, ac, ba, ca, cb, aba, acb, bac, cab, cba\}$	$\{\langle \varepsilon, \{\} \rangle, \langle a, \{\} \rangle, \langle b, \{\} \rangle, \langle c, \{\} \rangle, \langle aa, \{b\} \rangle, \langle ab, \{\} \rangle, \langle ac, \{\} \rangle, \langle ba, \{\} \rangle, \langle bb, \{a, c\} \rangle, \langle bc, \{a\} \rangle, \langle ca, \{\} \rangle, \langle cb, \{\} \rangle, \langle cc, \{a, b\} \rangle\}$
abca	$\{\varepsilon, a, b, c, ab, ac, ba, bc, ca, cb, aba, abc, acb, bac, bca, cab, cba\}$	$\{\langle \varepsilon, \{\} \rangle, \langle a, \{\} \rangle, \langle b, \{\} \rangle, \langle c, \{\} \rangle, \langle aa, \{b\} \rangle, \langle ab, \{\} \rangle, \langle ac, \{\} \rangle, \langle ba, \{\} \rangle, \langle bb, \{a, c\} \rangle, \langle bc, \{\} \rangle, \langle ca, \{\} \rangle, \langle cb, \{\} \rangle, \langle cc, \{a, b\} \rangle\}$
abbacc	$\{\varepsilon, a, b, c, ab, ac, ba, bb, bc, ca, cb, cc, aba, abb, abc, acb, acc, bac, bca, bba, cab, cba\}$	$\{\langle \varepsilon, \{\} \rangle, \langle a, \{\} \rangle, \langle b, \{\} \rangle, \langle c, \{\} \rangle, \langle aa, \{b\} \rangle, \langle ab, \{\} \rangle, \langle ac, \{\} \rangle, \langle ba, \{\} \rangle, \langle bb, \{\} \rangle, \langle bc, \{\} \rangle, \langle ca, \{\} \rangle, \langle cb, \{\} \rangle, \langle cc, \{\} \rangle\}$
baba	$\{\varepsilon, a, b, c, ab, ac, ba, bb, bc, ca, cb, cc, aba, abb, abc, acb, acc, bab, bac, bca, bba, cab, cba\}$	$\{\langle \varepsilon, \{\} \rangle, \langle a, \{\} \rangle, \langle b, \{\} \rangle, \langle c, \{\} \rangle, \langle aa, \{b\} \rangle, \langle ab, \{\} \rangle, \langle ac, \{\} \rangle, \langle ba, \{\} \rangle, \langle bb, \{\} \rangle, \langle bc, \{\} \rangle, \langle ca, \{\} \rangle, \langle cb, \{\} \rangle, \langle cc, \{\} \rangle\}$

Table 2: Progression of Lambert (2021)’s Online TSL learner over an handful of presented strings. The first row includes the empty grammar as initially assumed by the learning algorithm.

Subsequence	Intervener Sets
$\langle \rangle$	$\{\{\}\}$
$\langle ab \rangle$	$\{\{\}\}$
$\langle ac \rangle$	$\{\{\}\}$
$\langle ba \rangle$	$\{\{\}\}$
$\langle ca \rangle$	$\{\{\}\}$
$\langle cb \rangle$	$\{\{\}\}$
$\langle ab, ac \rangle$	$\{\{ba\}\}$
$\langle ab, ba \rangle$	$\{\{\}\}$
$\langle ab, cb \rangle$	$\{\{ac, ba\}\}$
$\langle ac, ba \rangle$	$\{\{cb\}\}$
$\langle ac, cb \rangle$	$\{\{\}\}$
$\langle ba, ac \rangle$	$\{\{\}\}$
$\langle ba, ba \rangle$	$\{\{ac, cb\}\}$
$\langle ba, cb \rangle$	$\{\{ac\}\}$
$\langle ca, ab \rangle$	$\{\{\}\}$
$\langle ca, ac \rangle$	$\{\{ab, ba\}\}$
$\langle ca, ba \rangle$	$\{\{ab\}\}$
$\langle ca, cb \rangle$	$\{\{ab, ac, ba\}\}$
$\langle cb, ba \rangle$	$\{\{\}\}$

Table 3: Augmented Subsequences extracted from cabacba by an ITSL learner with $k=2$ and $m=2$.

a width-2 string over the original alphabet (a width-2 substring of the input string), and thus we represent ITSL factors as tuples, with unary ITSL symbols separated by commas $\langle \sigma_1 \sigma_2, \sigma_3 \sigma_4, \dots \rangle$. Then, the set of attested augmented subsequences is as listed in Table 3. Finally, Table 4 exemplifies a run of the new ITSL learner on the same example strings as in Table 2.

In terms of space/time complexity, the original TSL learner total time complexity is $\mathcal{O}(n^k / (k-1)! \cdot |\Sigma| \log |\Sigma|)$, and its space complexity $\mathcal{O}\left(\binom{|\Sigma|}{|\Sigma|/2}\right)$. These results generalize to the ITSL learner, with an additional variable tied to the need of extracting subsequences and interveners

defined over complex input symbols. However, Lambert (2021) observes how the TSL learner as defined above can be thought of two separate learners run in parallel and that, thanks to free deletability of the non salient symbols, in most situations the left component of this composite grammar (G_l) is sufficient to both determine salience and function as an acceptor. Thus, an optimization is presented such that the TSL learner can converge in $\mathcal{O}(nk \log |\Sigma|)$ time and $\mathcal{O}(|\Sigma|^{k+1})$ space, where n is the number of strings to learn over, k is the width of the dependencies within the tier and $|\Sigma|$ is the size of the alphabet.

This optimization generalizes as is to ITSL, since nothing was changed in the structure of the learning procedure itself, and thus we only have to incorporate the additional complexity in deriving the salience of the contextually enriched “symbols”. Accordingly, the ITSL learner learns in $\mathcal{O}(nk \log(|\Sigma|^m))$ time and $\mathcal{O}((|\Sigma|^m)^{k+1})$ space, preserving the linear time and constant space requirements of the TSL version relative to the input size. Additionally, time and space complexity for the ITSL learner are exponential in the context width, and context and factor width, respectively.

5 Evaluating Online TSL and ITSL

The learning algorithm presented above offers formal convergence guarantees tied to the representation of ITSL and its impact on possible structural restrictions on the hypothesis space of the learner, assuming an input sample fully representative of the target language. In this last part of the paper, we offer a preliminary evaluation of the empirical performance of both the new ITSL learner and of Lambert (2021)’s TSL learner, in terms of consistency with the grammar generating the input (Aksënova, 2020). In particular,

w	G_ℓ	G_s
cabacba	$\{\langle \rangle, \langle ab \rangle, \langle ac \rangle, \langle ba \rangle, \langle ca \rangle, \langle cb \rangle, \langle ab, ba \rangle, \langle ac, cb \rangle, \langle ba, ac \rangle, \langle ca, ab \rangle, \langle cb, ba \rangle, \langle ab, ba, ac \rangle, \langle ac, cb, ba \rangle, \langle ba, ac, cb \rangle, \langle ca, ab, ba \rangle\}$	$\{\langle \rangle, \langle \rangle, \langle \rangle, \langle \langle ab \rangle, \langle \rangle\}, \langle \langle ac \rangle, \langle \rangle\}, \langle \langle ba \rangle, \langle \rangle\}, \langle \langle ca \rangle, \langle \rangle\}, \langle \langle cb \rangle, \langle \rangle\}, \langle \langle ab, ac \rangle, \langle \{ba\} \rangle\}, \langle \langle ab, ba \rangle, \langle \rangle\}, \langle \langle ab, cb \rangle, \langle \{ac, ba\} \rangle\}, \langle \langle ac, ba \rangle, \langle \{cb\} \rangle\}, \langle \langle ac, cb \rangle, \langle \rangle\}, \langle \langle ba, ac \rangle, \langle \rangle\}, \langle \langle ba, ba \rangle, \langle \{ac, cb\} \rangle\}, \langle \langle ba, cb \rangle, \langle \{ac\} \rangle\}, \langle \langle ca, ab \rangle, \langle \rangle\}, \langle \langle ca, ac \rangle, \langle \{ab, ba\} \rangle\}, \langle \langle ca, ba \rangle, \langle \{ab\} \rangle\}, \langle \langle ca, cb \rangle, \langle \{ab, ac, ba\} \rangle\}, \langle \langle cb, ba \rangle, \langle \rangle\}\}$
abca	$\{\langle \rangle, \langle ab \rangle, \langle ac \rangle, \langle ba \rangle, \langle bc \rangle, \langle ca \rangle, \langle cb \rangle, \langle ab, ba \rangle, \langle ab, bc \rangle, \langle ac, cb \rangle, \langle ba, ac \rangle, \langle bc, ca \rangle, \langle ca, ab \rangle, \langle cb, ba \rangle, \langle ab, ba, ac \rangle, \langle ab, bc, ca \rangle, \langle ac, cb, ba \rangle, \langle ba, ac, cb \rangle, \langle ca, ab, ba \rangle\}$	$\{\langle \rangle, \langle \rangle, \langle \langle ab \rangle, \langle \rangle\}, \langle \langle ac \rangle, \langle \rangle\}, \langle \langle ba \rangle, \langle \rangle\}, \langle \langle bc \rangle, \langle \rangle\}, \langle \langle ca \rangle, \langle \rangle\}, \langle \langle cb \rangle, \langle \rangle\}, \langle \langle ab, ac \rangle, \langle \{ba\} \rangle\}, \langle \langle ab, ba \rangle, \langle \rangle\}, \langle \langle ab, bc \rangle, \langle \rangle\}, \langle \langle ab, ca \rangle, \langle \{bc\} \rangle\}, \langle \langle ab, cb \rangle, \langle \{ac, ba\} \rangle\}, \langle \langle ac, ba \rangle, \langle \{cb\} \rangle\}, \langle \langle ac, cb \rangle, \langle \rangle\}, \langle \langle ba, ac \rangle, \langle \rangle\}, \langle \langle ba, ba \rangle, \langle \{ac, cb\} \rangle\}, \langle \langle ba, cb \rangle, \langle \{ac\} \rangle\}, \langle \langle bc, ca \rangle, \langle \rangle\}, \langle \langle ca, ab \rangle, \langle \rangle\}, \langle \langle ca, ac \rangle, \langle \{ab, ba\} \rangle\}, \langle \langle ca, ba \rangle, \langle \{ab\} \rangle\}, \langle \langle ca, cb \rangle, \langle \{ab, ac, ba\} \rangle\}, \langle \langle cb, ba \rangle, \langle \rangle\}\}$
abbacc	$\{\langle \rangle, \langle ab \rangle, \langle ac \rangle, \langle ba \rangle, \langle bb \rangle, \langle bc \rangle, \langle ca \rangle, \langle cb \rangle, \langle cc \rangle, \langle ab, ba \rangle, \langle ab, bb \rangle, \langle ab, bc \rangle, \langle ac, cb \rangle, \langle ac, cc \rangle, \langle ba, ac \rangle, \langle bb, ba \rangle, \langle bc, ca \rangle, \langle ca, ab \rangle, \langle cb, ba \rangle, \langle ab, ba, ac \rangle, \langle ab, bc, ca \rangle, \langle ac, cb, ba \rangle, \langle ba, ac, cb \rangle, \langle ca, ab, ba \rangle\}$	$\{\langle \rangle, \langle \rangle, \langle \langle ab \rangle, \langle \rangle\}, \langle \langle ac \rangle, \langle \rangle\}, \langle \langle ba \rangle, \langle \rangle\}, \langle \langle bb \rangle, \langle \rangle\}, \langle \langle bc \rangle, \langle \rangle\}, \langle \langle ca \rangle, \langle \rangle\}, \langle \langle cb \rangle, \langle \rangle\}, \langle \langle cc \rangle, \langle \rangle\}, \langle \langle ab, ac \rangle, \langle \{ba\} \rangle\}, \langle \langle ab, ba \rangle, \langle \rangle\}, \langle \langle ab, bb \rangle, \langle \rangle\}, \langle \langle ab, bc \rangle, \langle \rangle\}, \langle \langle ab, ca \rangle, \langle \{bc\} \rangle\}, \langle \langle ab, cb \rangle, \langle \{ac, ba\} \rangle\}, \langle \langle ab, cc \rangle, \langle \{ab, ac, ba, bb\} \rangle\}, \langle \langle ac, ba \rangle, \langle \{cb\} \rangle\}, \langle \langle ac, cb \rangle, \langle \rangle\}, \langle \langle ac, cc \rangle, \langle \rangle\}, \langle \langle ba, ac \rangle, \langle \rangle\}, \langle \langle ba, ba \rangle, \langle \{ac, cb\} \rangle\}, \langle \langle ba, cb \rangle, \langle \{ac\} \rangle\}, \langle \langle ba, cc \rangle, \langle \{ac\} \rangle\}, \langle \langle bb, ac \rangle, \langle \{ba\} \rangle\}, \langle \langle bb, ba \rangle, \langle \rangle\}, \langle \langle bb, cc \rangle, \langle \{ac, ba\} \rangle\}, \langle \langle bc, ca \rangle, \langle \rangle\}, \langle \langle ca, ab \rangle, \langle \rangle\}, \langle \langle ca, ac \rangle, \langle \{ab, ba\} \rangle\}, \langle \langle ca, ba \rangle, \langle \{ab\} \rangle\}, \langle \langle ca, cb \rangle, \langle \{ab\} \rangle\}, \langle \langle ca, cb \rangle, \langle \{ab, ac, ba\} \rangle\}, \langle \langle cb, ba \rangle, \langle \rangle\}\}$
baba	$\{\langle \rangle, \langle ab \rangle, \langle ac \rangle, \langle ba \rangle, \langle bb \rangle, \langle bc \rangle, \langle ca \rangle, \langle cb \rangle, \langle cc \rangle, \langle ab, ba \rangle, \langle ab, bb \rangle, \langle ab, bc \rangle, \langle ac, cb \rangle, \langle ac, cc \rangle, \langle ba, ab \rangle, \langle ba, ac \rangle, \langle bb, ba \rangle, \langle bc, ca \rangle, \langle ca, ab \rangle, \langle cb, ba \rangle, \langle ab, ba, ac \rangle, \langle ab, bc, ca \rangle, \langle ac, cb, ba \rangle, \langle ba, ac, cb \rangle, \langle ba, ab, ba \rangle, \langle ca, ab, ba \rangle\}$	$\{\langle \rangle, \langle \rangle, \langle \langle ab \rangle, \langle \rangle\}, \langle \langle ac \rangle, \langle \rangle\}, \langle \langle ba \rangle, \langle \rangle\}, \langle \langle bb \rangle, \langle \rangle\}, \langle \langle bc \rangle, \langle \rangle\}, \langle \langle ca \rangle, \langle \rangle\}, \langle \langle cb \rangle, \langle \rangle\}, \langle \langle cc \rangle, \langle \rangle\}, \langle \langle ab, ac \rangle, \langle \{ba\} \rangle\}, \langle \langle ab, ba \rangle, \langle \rangle\}, \langle \langle ab, bb \rangle, \langle \rangle\}, \langle \langle ab, bc \rangle, \langle \rangle\}, \langle \langle ab, ca \rangle, \langle \{bc\} \rangle\}, \langle \langle ab, cb \rangle, \langle \{ac, ba\} \rangle\}, \langle \langle ab, cc \rangle, \langle \{ab, ac, ba, bb\} \rangle\}, \langle \langle ac, ba \rangle, \langle \{cb\} \rangle\}, \langle \langle ac, cb \rangle, \langle \rangle\}, \langle \langle ac, cc \rangle, \langle \rangle\}, \langle \langle ba, ab \rangle, \langle \rangle\}, \langle \langle ba, ac \rangle, \langle \rangle\}, \langle \langle ba, ba \rangle, \langle \{ab\} \rangle\}, \langle \langle ba, ba \rangle, \langle \{ac, cb\} \rangle\}, \langle \langle ba, cb \rangle, \langle \{ac\} \rangle\}, \langle \langle ba, cc \rangle, \langle \{ac\} \rangle\}, \langle \langle bb, ac \rangle, \langle \{ba\} \rangle\}, \langle \langle bb, ba \rangle, \langle \rangle\}, \langle \langle bb, cc \rangle, \langle \{ac, ba\} \rangle\}, \langle \langle bc, ca \rangle, \langle \rangle\}, \langle \langle ca, ab \rangle, \langle \rangle\}, \langle \langle ca, ac \rangle, \langle \{ab, ba\} \rangle\}, \langle \langle ca, ba \rangle, \langle \{ab\} \rangle\}, \langle \langle ca, cb \rangle, \langle \{ab, ac, ba\} \rangle\}, \langle \langle cb, ba \rangle, \langle \rangle\}\}$

Table 4: Progression of the Online ITSL grammar over an handful of presented strings. The first row includes the empty grammar as initially assumed by the learning algorithm.

we implement the “two parallel learners” version of each algorithm as presented above in Python 3, both for the TSL learner and for our ITSL generalization.¹ We then evaluate performance when trained on input samples representative of patterns corresponding to various subregular classes and designed to mimic natural phonotactic phenomena.²

In particular, we conduct evaluations on 11 different training datasets, eight of which were artificially generated from a defined target grammar, and three were word-lists extracted from three natural language corpora with simplified alphabets

¹A Haskell implementation of the TSL learner is available as part of the Language Toolkit at <https://github.com/vvulpes0/Language-Toolkit-2>.

²Our code repository, with data for training and testing of both algorithms is available at https://github.com/jacobkj314/online_itsl.

(see Aksénova, 2020, for details). Specifically, our testing suite includes: word-final devoicing (strictly local); two vowel harmony patterns with a single constraint type (TSL); two vowel harmony patterns with multiple constraints to be evaluated over a single tier (TSL); and three types of ITSL patterns. For the ITSL dependencies, we consider an unbounded tone-plateauing pattern (Hyman and Katamba, 2010; Jardine, 2016) and a pattern of local dissimilation in which the tier consists of o, e, and a, where oe is a restricted bigram over the tier, but instances of o are only projected to the tier when followed by x.³ We also test a first-last harmony pattern, which

³This pattern was originally inspired by the ITSL analysis of Yaka nasal harmony (Hyman, 1995; Walker, 2000) as presented in De Santo and Aksénova (2021). Such analysis hinges on Yaka having nasal-stop clusters. A reviewer points out that it might

establishes a harmonic dependency between the first and the last element in the string. While this pattern has been argued to be unattested in natural languages (Lai, 2015; Avcu, 2017), it is a dependency worth testing in addition to the ones above, as it requires both elements in the constraint to be sensitive to their local context (the end and start symbols, respectively).

We train each learner on 1000 strings randomly sampled from the language generated by the target grammar for the artificial datasets, and on up to 130K words for the simplified natural language corpora (see Table 5). First, we set evaluation criteria defined according to the same pipeline as in Aks nova (2020), and comparable with the evaluation of the 2-ITSL₂ batch learner of De Santo and Aks nova (2021) as presented in Johnson and De Santo (2023). Embedding the learned grammar in an acceptor function, we filter strings from Σ^* in length-lexicographical order until 5000 strings are accepted. These 5000 strings are then additionally fed into an acceptor incorporating the original target grammar.⁴ Therefore, the score reported for each learner/target grammar pair in Table 6 indicates what proportion of the strings generated by the learned grammar were accepted by the target grammar. For the artificial languages, both learning and testing were repeated over 10 separate trials using a different set of input strings, and we report the average score over these 10 iterations of the full algorithm.

As shown in Table 6, both learners output highly consistent grammars for each of the SL and TSL patterns, even considering the relatively small input size. These results extend to the ITSL learner’s performance over the three ITSL patterns, fully consistently with theoretical expectations.⁵

Interestingly, the TSL learner shows (somewhat unexpected) differential performance on the ITSL data. As expected, this learner performs below or at chance for two of these patterns, but the consistency between learned and target grammars on the last ITSL pattern is strikingly high. Recall now that in an ITSL set-up, symbols are only relevant to the tier when conditioned by the appropriate local context. Thus, ITSL patterns viewed from a TSL perspective might look

be more appropriate to treat these not as sequences but as prenasalized stops and affricates, in which case the harmony pattern would simply be TSL. While getting the linguistic facts right is crucial for a subregular understanding of Yaka, for the sake of this paper what matters is that the abstract example is ITSL, and we keep it for comparison with Johnson and De Santo (2023).

⁴For the natural datasets, each acceptor function incorporates a grammar built to reproduce the underlying pattern, even if that grammar was not technically used to generate the input data.

⁵While omitted here because of space constraints, all learned grammars are available in the [repository](#) associated with this paper.

Mean Length (SD)	
Word-final devoicing	
A	10
N _G	14.90 (3.70)
Single vowel harmony without blocking	
A	10
N _F	13.92 (3.82)
Single vowel harmony with blocking	
A	10
Several vowel harmonies without blocking	
A	10
Several vowel harmonies with blocking	
A	7.32 (1.08)
N _T	7.85 (2.48)
Unbounded tone plateauing	
A	5
First-Last Assimilation	
A	10
Locally-driven long-distance assimilations (ITSL restriction)	
A	6.20 (0.93)

Table 5: Mean length of the strings in the datasets used for training the learners, based on the union of all sets of strings used by each trial. N_G: German; N_F: Finnish; N_T: Turkish. Where omitted, $SD=0$

relatively unconstrained: that is, no symbol evaluated in isolation might fit the no free deletion/insertion requirements needed to be considered salient for the tier. A preliminary qualitative evaluation of the output of the TSL learner over this pattern reveals that this is probably the reason for the high acceptance performance: the learner has converged to a strictly local grammar with no tier constraints.

Slightly in contrast with this observation though, the evaluation metric adopted above does not penalize strings that are accepted by the target grammar but rejected by the learned grammar, thus potentially favoring over-restricting grammars (i.e. under-generalization). As a preliminary investigation of this issue, we conduct a second batch of experiments. Table 7 shows, for all the artificial datasets, the proportion of the first 5000 strings accepted by the target grammar that are also accepted by the learned grammar. Together with the results of the previous experiment, these results support the intuition that the ITSL learner converges to more restrictive grammars than the TSL one, as it needs more data to infer that a segment is involved in a dependency independently of context. Still, the high performance of the TSL learner on the ITSL patterns, as well as the performance of both learners on TSL patterns with and without blocking deserve further attention. A more careful investigation of the learned grammars is needed to fully gain insights into the different

	TSL	ITSL
Word-final devoicing		
T	✓	✓
A	100%	100%
N _G	100%	100%
Single vowel harmony without blocking		
T	✓	✓
A	100%	100%
N _F	100%	100%
Single vowel harmony with blocking		
T	✓	✓
A	100%	100%
Several vowel harmonies without blocking		
T	✓	✓
A	100%	100%
Several vowel harmonies with blocking		
T	✓	✓
A	100%	100%
N _T	100%	100%
Unbounded tone plateauing		
T	✗	✓
A	9.97% (0.51%)	100%
First-Last Assimilation		
T	✗	✓
A	50.02%	100%
Locally-driven long-distance assimilation (ITSL restriction)		
T	✗	✓
A	94.88% (0.15%)	100%

Table 6: Results for Experiment 1. (T)heoretical expectations and performance as mean (and standard deviation) consistency (based on the first 5000 strings accepted by the learned grammar) of the grammars learned by Online TSL and Online ITSL learners on (A)rtificial and simplified (N)atural language input data-sets, measured over 10 iterations. N_G: German; N_F: Finnish; N_T: Turkish. Where omitted, $SD=0$.

performance of these learners. Understanding the kind of grammars more/less expressive learners converge onto when trained on theoretically less/more expressive patterns might also offer predictions for learnability expectations in human experiments.

6 Conclusion

Formal language theoretical insights have been argued to help bridge typological observations to learnability considerations (Lambert et al., 2021; De Santo and Rawski, 2022). While ITSL offers a good account of phonotactic dependencies from a descriptive characterization perspective, its overall relevance to this broader enterprise is limited by the implausibility of batch learning for humans. In this paper we presented a straightforward generalization of Lambert (2021)’s TSL incremental learner to ITSL, leveraging a more complex definition of tier-symbols in order

	TSL	ITSL
Word-final devoicing		
A	99.96%	71.22% (2.64%)
Single vowel harmony without blocking		
A	9.24%	7.78%
Single vowel harmony with blocking		
A	86.54%	18.64% (1.25%)
Several vowel harmonies without blocking		
A	12.64%	10.26%
Several vowel harmonies with blocking		
A	99.82%	56.90% (1.53%)
Unbounded tone plateauing		
A	99.96%	99.86%
First-Last Assimilation		
A	78.14%	73.01% (0.81%)
Locally-driven long-distance assimilation (ITSL restriction)		
A	99.96%	59.79% (1.23%)

Table 7: Results for Experiment 2. Performance as mean (and standard deviation) completeness (based on the first 5000 strings accepted by the target grammar) of the grammars learned by Online TSL and Online ITSL learners on Artificial language input data-sets, measured over 10 iterations. Where omitted, $SD=0$.

to determine salience. Taking into account the additional complexity brought by moving segments from unigrams to m -grams, this learner maintains the complexity constraints of the original TSL learner, and its convergence guarantees. An evaluation of learning performance over a variety of patterns also demonstrates the viability of this learning approach beyond theoretical guarantees. Moreover, as already suggested by Johnson and De Santo (2023), we argue that implemented grammatical inference algorithms allow to probe the information about target patterns present in phonotactic corpora, facilitating the study of the relation between data and learnability in humans and machines. In the future, it would be interesting to explore the extent to which this approach can be used to extend Lambert (2021)’s learner to the input-output languages of Graf and Mayer (2018), to stochastic counterparts to TSL and ITSL (Mayer, 2021), and to multiple independent TSL constraints (De Santo and Graf, 2019; McMullin et al., 2019; De Santo and Aks nova, 2021).

Acknowledgements

We are grateful to SCiL’s anonymous reviewers for their valuable feedback on this manuscript.

References

Al na Aks nova. 2020. *Tool-assisted induction of subregular languages and mappings*. Ph.D. thesis, State University of New York at Stony Brook.

- Alëna Aksënova, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of SIGMorPhon 2016*.
- Richard B. Applegate. 1972. *Ineseno Chumash grammar*. Ph.D. thesis, UC Berkeley.
- Enes Avcu. 2017. Experimental investigation of the subregular hierarchy. In *Proceedings of the 35th West Coast Conference on Formal Linguistics at Calgary, Alberta Canada*.
- Hyunah Baek. 2017. Computational representation of unbounded stress patterns: tiers with structural features. In *Proceedings of the 53rd Meeting of the Chicago Linguistic Society (CLS53)*.
- Jane Chandlee and Jeffrey Heinz. 2016. Computational phonology. Ms., Haverford College and University of Delaware.
- Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry*, 49:23–60.
- Aniello De Santo and Alëna Aksënova. 2021. Learning interactions of local and non-local phonotactic constraints from positive input. *Proceedings of the Society for Computation in Linguistics*, 4(1):167–176.
- Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *International conference on formal grammar*, pages 35–50. Springer.
- Aniello De Santo and Jonathan Rawski. 2022. Mathematical linguistics and cognitive complexity. In *Handbook of Cognitive Mathematics*, pages 1–38. Springer.
- E Mark Gold. 1967. Language identification in the limit. *Information and control*, 10(5).
- John Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, MIT, Cambridge, MA.
- Thomas Graf. 2017. [The power of locality domains in phonology](#). *Phonology*, 34:385–405.
- Thomas Graf. 2022a. Subregular linguistics: bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3-4):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.
- Thomas Graf and Connor Mayer. 2018. Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of SIGMorPhon 2018*.
- Jeffrey Heinz. 2010. [String extension learning](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906.
- Jeffrey Heinz. 2011a. Computational phonology – part 1: Foundations. *Language and Linguistics Compass*, 5(4):140–152.
- Jeffrey Heinz. 2011b. Computational phonology – part 2: Grammars, learning, and the future. *Language and Linguistics Compass*, 5(4):153–168.
- 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.
- Larry M Hyman. 1995. Nasal consonant harmony at a distance the case of yaka. *Studies in African Linguistics*, 24(1):6–30.
- Larry M Hyman and Francis X Katamba. 2010. Tone, syntax, and prosodic domains in luganda. *ZAS Papers in Linguistics*, 53:69–98.
- Adam Jardine. 2016. [Computationally, tone is different](#). *Phonology*.
- Adam Jardine and Jeffrey Heinz. 2016. [Learning tier-based strictly 2-local languages](#). *Transactions of the ACL*, 4:87–98.
- Adam Jardine and Kevin McMullin. 2017. Efficient learning of tier-based strictly k -local languages. In *Language and Automata Theory and Applications, 11th International Conference*, LNCS, pages 64–76. Springer.
- Jacob K Johnson and Aniello De Santo. 2023. Evaluating a phonotactic learner for MITS L -(2, 2) languages. *Proceedings of the Society for Computation in Linguistics*, 6(1):379–382.
- Regine Lai. 2015. Learnable vs. unlearnable harmony patterns. *LI*, 46(3):425–451.
- Dakotah Lambert. 2021. [Grammar interpretations and learning tsl online](#). In *Proceedings of the Fifteenth International Conference on Grammatical Inference*, volume 153 of *Proceedings of Machine Learning Research*, pages 81–91. PMLR.
- Dakotah Lambert, Jonathan Rawski, and Jeffrey Heinz. 2021. Typology emerges from simplicity in representations and learning. *Journal of Language Modelling*, 9.
- Dakotah Lambert and James Rogers. 2020. Tier-based strictly local stringsets: Perspectives from model and automata theory. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 159–166.
- 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.
- Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from Uyghur backness harmony. In *Formal Grammar 2018. Lecture Notes in Computer Science*, vol. 10950, pages 62–83. Springer, Berlin, Heidelberg.
- Kevin McMullin. 2016. [Tier-based locality in long-distance phonotactics?: learnability and typology](#). Ph.D. thesis, U. of British Columbia.

- Kevin McMullin, Alëna Aksënova, and Aniello De Santo. 2019. [Learning phonotactic restrictions on multiple tiers](#). *Proceedings of SCiL 2019*, 2(1):377–378.
- Robert McNaughton and Seymour Papert. 1971. *Counter-Free Automata*. MIT Press, Cambridge.
- 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.
- Imre Simon. 1975. [Piecewise testable events](#). In *Automata Theory and Formal Languages 2nd GI Conference*, volume 33 of *Lectures Notes in Computer Science*, pages 214–222, Berlin. Springer.
- Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of SCiL 2019*, pages 267–276.
- Rachel Walker. 2000. [Yaka nasal harmony: Spreading or segmental correspondence?](#) *Annual Meeting of the Berkeley Linguistics Society*, 26(1):321–332.

Neural language model gradients predict event-related brain potentials

Stefan L. Frank

Centre for Language Studies
Radboud University
Nijmegen, the Netherlands
stefan.frank@ru.nl

Abstract

Fitz and Chang (2019) argue that event-related brain potentials during sentence comprehension result from the detection and incorporation of word-prediction error. Specifically, the N400 component would correlate with prediction error while the P600 component would be indicative of error backpropagation in the language system. The current work evaluates this hypothesis on a corpus of EEG data recorded during naturalistic sentence reading. Word-prediction error and backpropagated error were estimated by an LSTM language model that processed the same 205 English sentences as the human participants. At each word, the word's surprisal and the total gradient of recurrent-layer connections were collected for comparison to the sizes of the N400 and P600 components. Consistent with the theory, higher surprisal resulted in stronger N400 while higher gradient resulted in stronger P600, and ERPs on content words were more sensitive to surprisal whereas ERPs on function words were more sensitive to gradient. However, a detailed analysis of the neural signal's time course indicates that the apparent P600 effect could be interpreted as a reversed N400 effect.

1 Introduction

1.1 Event-related brain potentials

When people engage in language comprehension, their brains display particular patterns of electrical activity, a small part of which can be picked up by electrodes on the scalp. This method, known as electroencephalography (EEG), has revealed several typical deflections in measured voltage in response to word perception. These deflections are known as event-related brain potentials (ERPs) and particular ERP components can be identified by their timing and scalp distribution.

Arguably the two most studied components are the N400 and the P600. The first of these components is a negative-going voltage deflection that

peaks at around 400 ms after word onset, hence the name N400. The second ERP component goes in the positive direction and peaks at around 600 ms after word onset, hence the name P600.

The N400 is well known to be stronger (i.e., more negative) on words that are syntactically correct but semantically odd (Kutas and Hillyard, 1980), or simply have lower occurrence probability as estimated by human judgements (Kutas and Hillyard, 1984) or language models (Frank et al., 2015). A stronger P600 was originally thought to be indicative of syntactic violations and anomalies (Osterhout and Holcomb, 1992) but has also been found in different types of well-formed sentences, for example in response to a word that completes a long-distance dependency (Kaan et al., 2000) or is used ironically (Regel et al., 2014).

1.2 Models of the N400 and P600

Several computational models have been proposed as explanations of N400 and P600 effects in language comprehension (Brouwer et al., 2017, 2021; Fitz and Chang, 2019; Li and Futrell, 2022, 2023). These models agree that the N400 is stronger in response to a word that was less expected, although they differ in how this prediction error is quantified. As for the P600, all models assume that its size correlates with the extent to which the incoming word results in an update of some representation, but they disagree on the content of this representation.

According to the Retrieval-Integration account by Brouwer et al. (2017, 2021), the P600 corresponds to the update of a representation of the situation described by the sentence or text, which in their model is represented at the output layer of a recurrent neural network. In contrast, the model by Li and Futrell (2022, 2023) assumes that the P600 reflects the update in the reader's (or listener's) beliefs about the word sequence processed so far. Finally, Fitz and Chang's (2019) Error Propagation account claims that processing a word can lead

to an update of language knowledge, that is, to learning about the language's statistics or syntactic patterns. The P600 would reflect the size of this knowledge update, which can be quantified as the backpropagated word-prediction error in a neural network.

Fitz and Chang (2019) tested their theory in the Dual-Path model (Chang et al., 2006), a recurrent neural network (RNN) that differs from most language models in that it splits processing into two paths: a syntactic path that takes care of word ordering and a semantic path that maps propositional meaning onto sentences. During model training, each word's prediction error backpropagates through both paths but converges on a single recurrent layer. Fitz and Chang (2019) take the summed absolute gradients of recurrent-layer connection weights as their predictor of the P600 induced by the word, and show that this accounts for many results from human P600 experiments, such as the stronger P600 response to syntactic violations (compared to grammatically correct alternatives) caused by subject-verb number disagreement or incorrect verb-tense inflections. More recently, Verwijmeren et al. (2023) demonstrated that the Error Propagation account, implemented in a bilingual version of the Dual-Path model (Tsoukala et al., 2021) can explain why subject-verb number disagreement results in an enhanced N400 in beginning second-language learners but an enhanced P600 in more advanced learners.

1.3 The current study

In spite of its successes, evaluation of the Error Propagation account has been hampered by the limitations of training the Dual-Path model, which requires each sentence to be paired with its propositional semantics. In practice, this means that the model can only be trained on artificial, toy versions of real languages. Although this often suffices for investigating specific psycholinguistic phenomena, it makes broad-coverage validation on natural language impossible. Crucially, Fitz and Chang (2019) did also investigate the Dual-Path model's ERP predictions when the semantic path was removed, in effect reducing it to a normal simple recurrent network (Elman, 1990) trained on the same toy language as the full Dual-Path model. Results were similar to those of the full model (at least, as far as the P600 was concerned) suggesting that the semantic path contributed little, if anything, to the P600 prediction.

If semantic knowledge is indeed not required to explain P600 effects, the Error Propagation account can also be evaluated in a way that is more similar to common practice in computational linguistics: train a neural language model on a natural language corpus and then test it on a novel sample of sentences. This is exactly the approach I take here. An RNN is used to estimate word surprisal and the word-induced gradients of recurrent-layer connection weights, at each word of English sentences that were also read by native English speakers while their EEG was recorded. Next, linear regression predicts the human N400 and P600 sizes from the model-derived surprisal and gradient values.

The results of the current study show that, as predicted by the Error Propagation account, higher surprisal correlates with stronger N400 while higher gradient correlates with stronger P600. Unexpectedly, however, higher surprisal and gradient also correspond to *weaker* P600 and N400, respectively. This suggests that the two predictors in fact have the same effect on the EEG signal (albeit in opposite directions) and the apparent separable effects on the N400 and P600 components are an artifact caused by their spatiotemporal overlap. This interpretation is supported by additional regression analyses: Across time and scalp locations, surprisal and gradient show similar effects on the EEG signal. Hence, backpropagated word-prediction error may thus correspond to weaker N400s as opposed to stronger P600s.

2 Methods

2.1 EEG data

Frank et al. (2015) published EEG data recorded on 32 electrodes, from 24 native English speakers reading 205 English sentences that were extracted from novels. The sentences were presented word-by-word¹ at a fixed location to minimize eye movements that interfere with the EEG signal. The duration between consecutive word onsets was at least 627 ms and increased by 20 ms per character, that is, it was word-length dependent.

Time-locked to each word onset, the EEG signals were averaged over different combinations of scalp electrodes and time windows to obtain six ERP components that have been investigated in the psycho- and neurolinguistic literature (see Frank et al., 2015, for details). The baseline level for each

¹Punctuation marks were attached to the preceding word and contractions were presented as single words.

component was the average over that component's electrodes during the 100 ms leading up to word onset. Here, I investigate only the N400 and P600 components. The N400 is defined as the average voltage from 300 to 500 ms after word onset, over 12 centro-parietal electrodes. The P600 is the average from 500 to 700 ms after word onset, over 18 electrodes that include the N400 electrodes but also more temporally located ones.

2.2 Language model²

2.2.1 Model architecture

As mentioned in Section 1.2, [Fitz and Chang \(2019\)](#) tested their theory in an RNN next-word prediction model that has both a semantic and a syntactic pathway (although they found the semantic pathway not to be critical to the P600 predictions). In order to stay as close as possible to that architecture while allowing it to be trained on a natural language corpus, I sacrificed the semantic pathway, leaving a plain, single-layer RNN; more specifically, a Long Short-Term Memory model (LSTM; [Hochreiter and Schmidhuber, 1997](#)) with 400-dimensional input embeddings and a 500-unit recurrent (LSTM) layer followed by a 400-unit hidden layer before the softmax output layer.

2.2.2 Model training

Training sentences were extracted from the first 7 slices of the ENCOW16 corpus of English sentences from the web ([Schäfer, 2015](#)). First, a vocabulary was created comprising the 20,000 most frequent tokens in the first slice of ENCOW16 plus all tokens from the 205 experimental stimuli sentences.³ Next, all sentences were selected that contain only vocabulary tokens and are no less than 3 and no more than 50 tokens in length. This resulted in a total of just under 81.6M training sentences with over 1.4B tokens of 21,918 types. All tokens from the experimental stimuli were attested in this training set. The training set was presented to the network for 1 training epoch.

²The language model's PyTorch ([Paszke et al., 2019](#)) code, training data, and trained models can be downloaded from <https://osf.io/a6g4f>

³Sentences from another psycholinguistic study were also included but these are irrelevant to the current work. The corpus sentence tokenization was adapted to that of the EEG experiment by merging the parts of a contraction (e.g., the two corpus tokens "do_n't" become the single token "don't"). Punctuation marks remained individual tokens.

2.2.3 Model testing

At several points during training, the LSTM processed the 205 sentences from the [Frank et al. \(2015\)](#) EEG study and estimated each word's surprisal ([Hale, 2001](#); [Levy, 2008](#)), that is, the negative log-probability of the word conditioned on the sentence so far. Surprisal values quantify word-prediction error and are expected to correlate with the size of the N400 component, as was already shown by [Frank et al. \(2015\)](#) on the same EEG data but using surprisal estimates from much smaller language models.

Each word's prediction error is backpropagated through the network ([Rumelhart et al., 1986](#)) resulting in a gradient for each connection weight. Following [Fitz and Chang \(2019\)](#), I take the summed absolute values of the gradients in the recurrent layer; an aggregate measure I simply refer to as 'the gradient'. Unlike [Fitz and Chang's \(2019\)](#) simple recurrent network's units, LSTM units have four types of connection (for the memory cell, and the input, output, and forget gates). The gradient measure is computed over all these weights together. Note that the gradients are computed for the 205 experimental sentences but not actually applied during model testing, that is, the connection weights are not updated.

2.3 Data analysis

Following [Frank et al. \(2015\)](#), I exclude from analysis all sentence-initial words, words attached to punctuation, and any data point from part of the EEG signal that was considered an artifact (mostly due to eye blinks). This left a total of 33,476 data points (i.e., combinations of participants and word tokens) for analysis. Statistical models were fit by the MixedModels package ([Bates et al., 2023](#)) in Julia ([Bezanson et al., 2017](#)).⁴

2.3.1 Standard ERP analysis

Separate sets of linear mixed-effects regression analyses were run with N400 size or P600 size as the dependent variable. Both analyses included surprisal and gradient as predictors, and the following covariates of no interest: the component's baseline, the position of the sentence in the experiment session, the position of the word in the sentence, the log-transformed frequency of the word in the British National Corpus, and the word's length

⁴Analysis code and EEG data can be downloaded from <https://osf.io/a6g4f>.

(number of characters). All predictors were standardized. The regression models included a by-token random intercept and slope of sentence position, and a by-participant random intercept and slopes of surprisal, gradient, sentence position, word position, log-transformed word frequency, and word length.

I take the t -statistics of surprisal and gradient as measures of the extent to which they are predictive of ERP size. A negative t -value of surprisal is expected in the N400 analysis (higher surprisal leads to a stronger, i.e., more negative-going N400) and a positive t -value of gradient in the P600 analysis (higher gradient leads to a stronger, i.e., more positive-going P600). When $|t| > 2$, this roughly corresponds to an effect that is statistically significant with $p < .05$.

2.3.2 Regression ERP analysis

A follow-up analysis does not take the ERP sizes as dependent variables but follows the ‘regression ERP’ (rERP) approach of [Smith and Kutas \(2015\)](#). This comes down to fitting a regression model to the set of EEG samples at each time point (relative to word onset) and electrode, and then plotting the coefficients of the predictors of interest as if they are ERP curves. All these regression models have both surprisal and gradient as predictors, with the same covariates and random-effect structure as in the standard ERP analysis discussed above. To reduce computation time, this analysis is only performed for the 7 most central electrodes, using only the fully trained network’s surprisal and gradient estimates.

3 Results

3.1 Surprisal and gradient measures

Figure 1 shows how the per-sentence averages of surprisal and gradient, as well as the correlation between them, change over network training. As expected, surprisal decreases with more training, indicating the the network makes increasingly accurate next-word predictions. Put differently: it is learning the statistical patterns of English.

Perhaps more surprisingly, gradient initially remains low, so not much of the prediction error in the output units results in changes in the recurrent connection weights. After approximately 100K training sentences, prediction error is increasingly backpropagated to the LSTM layer until the gradient more or less stabilizes after 10M sentences.

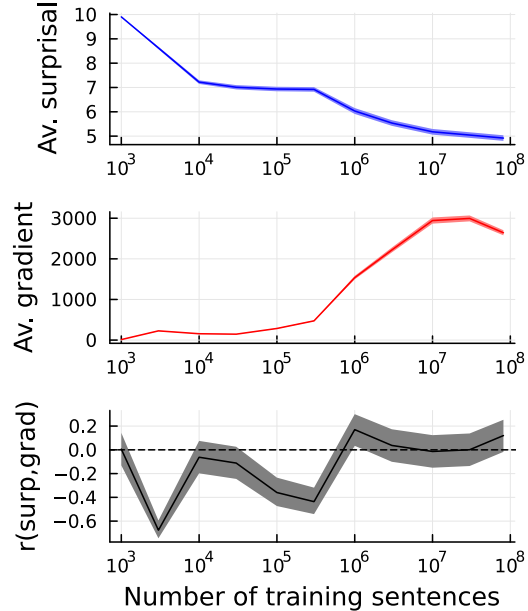


Figure 1: Average surprisal (top), average gradient (center), and their correlation (bottom) as a function of the number of training sentences. Shaded areas indicate 95% confidence intervals. Averages, correlations, and confidence intervals are computed over the 205 per-sentence averages because within a sentence, the word-level values do not constitute independent measurements.

There is a medium-sized, negative correlation between surprisal and gradient until about 300K training sentences, but after the network has been trained on 1M sentences the correlation is no longer statistically significant. The negative correlation early in training may seem hard to reconcile with the fact that output prediction error (quantified by surprisal) is backpropagated and then forms the driving force behind connection weight update (quantified by gradient). I return to this issue in Section 4.2.

3.2 Standard ERP analysis

Figure 2 shows how the fit of surprisal and gradient to ERP size changes as the number of training sentences increases. Clearly, high surprisal leads to a stronger (more negative-going) N400, and this effect of surprisal increases as the model is more thoroughly trained. The effect of gradient on P600 size is weaker, but it is in the positive direction and also increases with more training.

N400 effects are known to be mostly driven by content words ([Frank et al., 2015](#)) while the P600 has often been associated with syntactic processing.

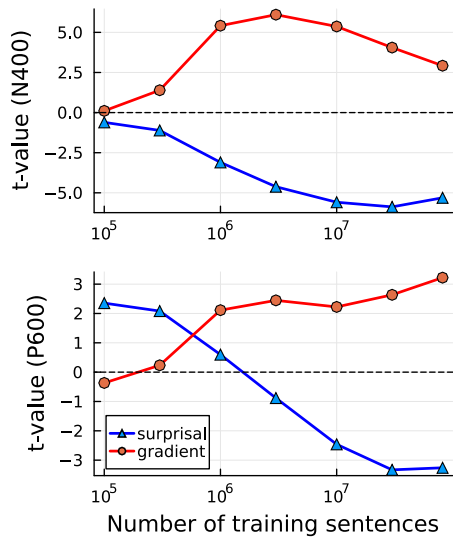


Figure 2: t -statistics for the effects of surprisal (blue triangles) and gradient (red circles) on N400 size (top) and P600 size (bottom), as a function of the number of training sentences.

To investigate if this distinction is apparent in the effects of surprisal and gradient, content and function words were also analyzed separately.⁵ As Figure 3 shows, surprisal is more predictive of ERP size on content words than on function words, whereas the same is not the case for gradient.

3.3 Regression ERP analysis

In addition to the expected effects of surprisal and gradient, the standard ERP analysis of Section 3.2 revealed that higher surprisal results in weaker P600 and that higher gradient results in weaker N400 (although the latter effect decreases after about 3M training sentences). This is most likely due to spatio-temporal overlap between the two ERP components (Brouwer and Crocker, 2017), which raises the question whether the apparent P600 effect of gradient truly is a P600 or if it could be a reversed effect on the N400 that only looks like a P600 because the two components are not fully separated in time and electrode location.

The results of the rERP analysis in Figure 4 suggest that this is indeed the case: The positive effect of gradient peaks at around 400 ms instead of 600 ms after word onset.

⁵This follows the content/function-word split provided by Frank et al. (2015), where 53.2% of words were designated as content words and 46.8% as function words. Contractions were excluded.

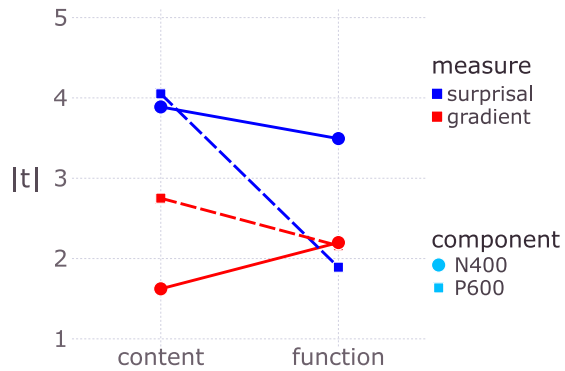


Figure 3: Absolute values of the t -statistics for the effects of surprisal (blue) and gradient (red), after training on the full dataset, analyzed separately for content and function words. Solid lines and round markers denote N400 effects; dashed lines and square markers denote P600 effects.

4 Discussion

4.1 The Error Propagation account

According to the Error Propagation account of language-related ERPs, the N400 during sentence comprehension reflects word-prediction error and the P600 corresponds to the (potential) update in language knowledge caused by word processing. Fitz and Chang (2019) quantify the size of this update in terms of the gradients of recurrent connection weights in an RNN. So far, this hypothesis had only been evaluated by comparing P600-size predictions between pairs of input sentences that constituted ‘toy’, artificial versions of controlled stimuli from psycholinguistics experiments. The current study, in contrast, is the first to validate the Error Propagation account on EEG data from a naturalistic sentence comprehension experiment, extracting the N400 and P600 predictions from a neural language model trained on a reasonably sized corpus of natural language text.

The results partially support Fitz and Chang’s (2019) theory: prediction error (surprisal) is predictive of the N400 and backpropagated error (gradient) corresponds to a positive-going ERP. Also, the finding that only surprisal effects are stronger for content words than for function words (Figure 3) is consistent with the idea that surprisal mainly affects the N400 and gradient the P600. Clearly, surprisal and gradient have separable effects in the expected directions. However, the regression-ERP analysis revealed that what was assumed to be an P600 in fact has a time course that is more like that of an N400 (one that is weaker for higher gradient)

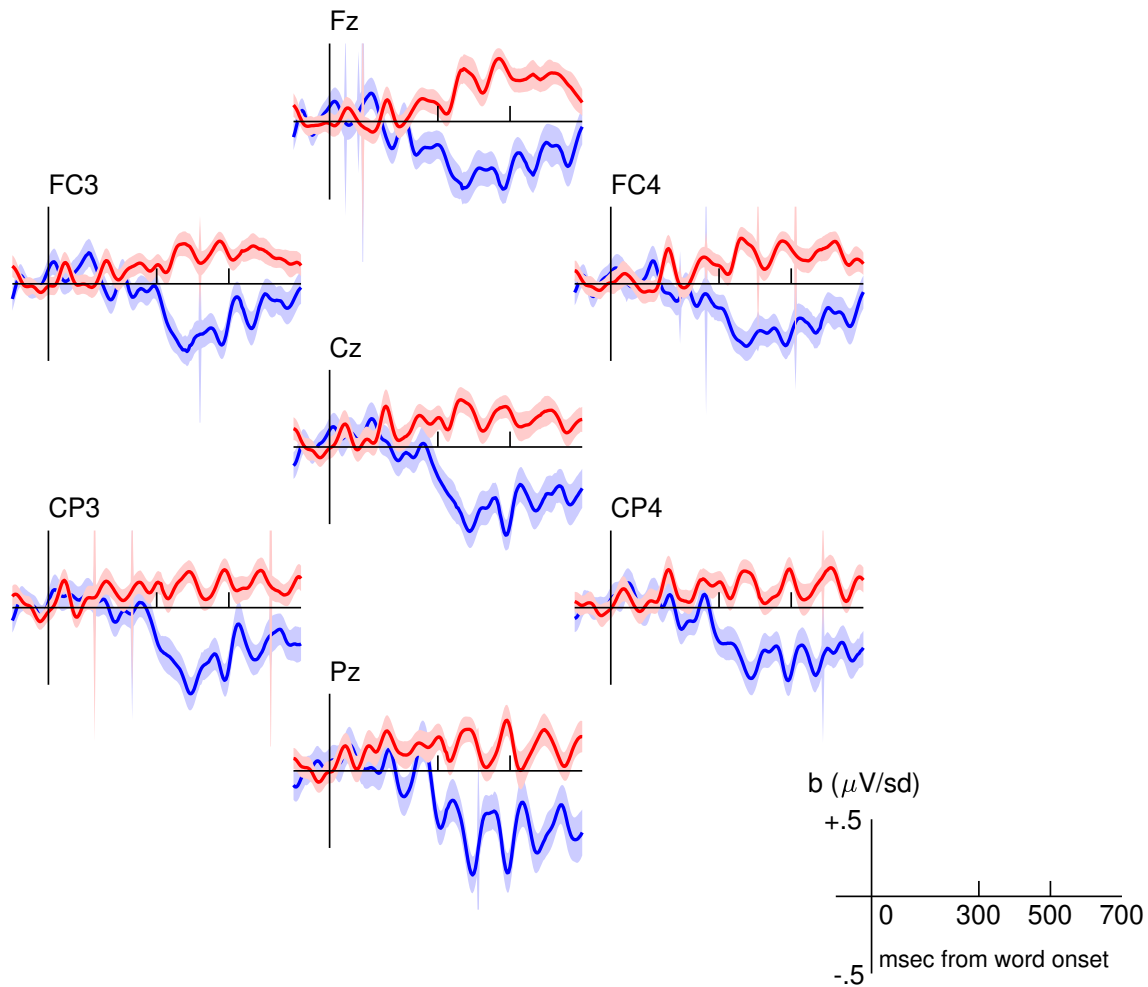


Figure 4: Topographic map of rERP curves. Each plot corresponds to one electrode and the curves show the effects (regression coefficients) of surprisal (blue) or gradient (red) on voltage at the electrode, time-locked to word onset. Shaded areas indicate standard errors.

and may therefore not be a true P600 ERP component. To summarize, the findings are inconclusive: There is an effect of gradient although it may not be exactly the effect predicted by the theory. The question remains whether surprisal and gradient indeed form qualitatively different linking hypothesis between properties of the language model and properties of the EEG signal, or if there are merely two sides of the same coin, with gradient modulating (i.e., weakening) the effect of surprisal on the N400.

4.2 Correlation between surprisal and gradient

Figure 1 revealed an unexpected and fairly large negative correlation between surprisal and gradient during early stages of network training. Although error backpropagation can only occur to the extent

that there is prediction error, gradient and surprisal are not simply the same measure because the gradient of a connection's weight also depends on the activation going into that connection. Moreover, there can be confounding variables between surprisal and gradient. Possibly, a confound with word frequency is responsible for the observed negative correlation between surprisal and gradient. Word frequencies will be among the first statistics learned by the network, where they are encoded in the output units' biases. As is visible from Figure 1, at the point in training when surprisal and gradient are negatively correlated, average surprisal has dropped but gradient remains close to 0, indicating that very little prediction error is backpropagated: learning mostly takes place at the output connections and biases. Presumably, the output units representing high-frequency words are the first to have

fairly stable biases, so prediction errors on these words are the first to be backpropagated, resulting in non-zero gradients in the LSTM layer. Meanwhile, prediction error on low-frequency words still mostly leads to changes in output biases. As a consequence, gradients in the LSTM layer will be higher on higher-frequency (and, consequently, lower-surprisal) words, that is, the surprisal and gradient measures are negatively correlated.

4.3 Evaluation on experimental versus naturalistic items

P600 effects in sentence comprehension are mostly, if not exclusively, investigated on sentences that result in comprehension difficulty, be it due to (morpho)syntactic violations (Coulson et al., 1998), garden-path structures (Osterhout and Holcomb, 1992), long-distance dependencies (Kaan et al., 2000), or semantic incongruity (Kuperberg et al., 2003). The same is true for all models of ERP effects discussed in Section 1.2. In contrast, the Frank et al. (2015) test sentences were sampled from novels and are therefore not expected (nor manipulated) to evoke any specific difficulty. It is not impossible that for such easy-to-understand sentences, the P600 occurs earlier, coinciding with the N400. Future research may reveal if the Error Propagation account, in combination with a neural language model trained on natural text, predicts more standard P600 effects on the hand-crafted sentences from psycholinguistic experiments.

Note that such an evaluation on realistic data is not possible with the Retrieval-Integration account (Brouwer et al., 2017, 2021) because that account takes the P600 to reflect the update of a representation of the described situation, and therefore requires such a representation – something that is not easily formalized for natural language. In contrast, the Li and Futrell (2022, 2023) model only requires knowledge of syntactic word-order patterns and therefore can be (and, in fact, has been) evaluated using the actual stimuli of psycholinguistic experiments.

4.4 Improving the language model

Another potential avenue for future research is to investigate whether improving the quality of the language model also improves its ERP predictions. The current work stayed as close as possible to that of Fitz and Chang (2019), using a single-layer RNN. Increasing the network’s size (e.g., adding layers), changing the architecture (e.g., a Trans-

former instead of an LSTM), and increasing the amount of training data will certainly result in a more accurate language model. In general, better language models more accurately fit human processing measures, be it from EEG, eye tracking, or fMRI (Merkx and Frank, 2021; Schrimpf et al., 2021). With multiple network layers to extract the gradient measure from, it may also be possible to distinguish between P600s resulting from different aspects of language processing.

5 Conclusion

This study tested Fitz and Chang’s (2019) Error Propagation account of event-related brain potentials during sentence comprehension, by extracting N400 and P600 predictions from a neural language model that processed the same sentences as humans in an EEG study. In line with the theory, the model’s word-prediction error (surprisal) correlated with N400 size. Backpropagated word-prediction error, which quantifies the potential update of the reader’s language knowledge, is measurable in the EEG signal but it remains unclear whether this takes the form of a stronger P600 or a weaker N400.

Acknowledgements

I am grateful to Naomi Shapiro for helpful comments on an earlier version of this paper.

References

- Douglas Bates, Phillip Alday, Dave Kleinschmidt, José Bayoán Santiago Calderón, Likán Zhan, Andreas Noack, Milan Bouchet-Valat, Alex Arslan, Tony Kelman, Antoine Baldassari, Benedikt Ehinger, Daniel Karrasch, Elliot Saba, Jacob Quinn, Michael Hatherly, Morten Piibeleht, Patrick Kofod Mogensen, Simon Babayan, Tim Holy, Yakir Luc Gagnon, and Yoni Nazarathy. 2023. [Juliastats/mixedmodels.jl: v4.22.3](https://juliastats.com/mixedmodels.jl/v4.22.3).
- Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. [Julia: A fresh approach to numerical computing](https://www.siam.org/publications/siam-reviews). *SIAM review*, 59(1):65–98.
- Harm Brouwer and Matthew W. Crocker. 2017. [On the proper treatment of the N400 and P600 in language comprehension](https://doi.org/10.3389/fpsyg.2017.01327). *Frontiers in Psychology*, 8:1327.
- Harm Brouwer, Matthew W. Crocker, Noortje J. Venhuizen, and John C.J. Hoeks. 2017. [A neurocomputational model of the N400 and the P600 in language processing](https://doi.org/10.1177/0956797617708135). *Cognitive Science*, 41:1318–1352.

- Harm Brouwer, Francesca Delogu, Noortje J. Venhuizen, and Matthew W. Crocker. 2021. [Neurobehavioral correlates of surprisal in language comprehension: A neurocomputational model](#). *Frontiers in Psychology*, 12:615538.
- Franklin Chang, Gary S. Dell, and Kathryn Bock. 2006. [Becoming syntactic](#). *Psychological Review*, 113(2):234.
- Seana Coulson, Jonathan W. King, and Marta Kutas. 1998. [Expect the unexpected: Event-related brain response to morphosyntactic violations](#). *Language and Cognitive Processes*, 13(1):21–58.
- J. L. Elman. 1990. [Finding structure in time](#). *Cognitive Science*, 14:179–211.
- Hartmut Fitz and Franklin Chang. 2019. [Language ERPs reflect learning through prediction error propagation](#). *Cognitive Psychology*, 111:15–52.
- Stefan L. Frank, Leun J. Otten, Giulia Galli, and Gabriella Vigliocco. 2015. [The ERP response to the amount of information conveyed by words in sentences](#). *Brain and Language*, 140:1–11.
- John T. Hale. 2001. [A probabilistic Early parser as a psycholinguistic model](#). In *Proceedings of the second conference of the North American chapter of the Association for Computational Linguistics*, volume 2, pages 159–166. Pittsburgh, PA: Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Computation*, 9(8):1735–1780.
- Edith Kaan, Anthony Harris, Edward Gibson, and Phillip Holcomb. 2000. [The P600 as an index of syntactic integration difficulty](#). *Language and Cognitive Processes*, 15(2):159–201.
- Gina R. Kuperberg, Tatiana Sitnikova, David Caplan, and Phillip J. Holcomb. 2003. [Electrophysiological distinctions in processing conceptual relationships within simple sentences](#). *Cognitive Brain Research*, 17(1):117–129.
- Marta Kutas and Steven A. Hillyard. 1980. [Reading senseless sentences: Brain potentials reflect semantic incongruity](#). *Science*, 207:203–205.
- Marta Kutas and Steven A. Hillyard. 1984. [Brain potentials during reading reflect word expectancy and semantic association](#). *Nature*, 307:161–163.
- Roger Levy. 2008. [Expectation-based syntactic comprehension](#). *Cognition*, 106:1126–1177.
- Jiaxuan Li and Richard Futrell. 2022. [A unified information-theoretic model of EEG signatures of human language processing](#). In *NeurIPS 2022 Workshop on Information-Theoretic Principles in Cognitive Systems*.
- Jiaxuan Li and Richard Futrell. 2023. [A decomposition of surprisal tracks the N400 and P600 brain potentials](#). In *Proceedings of the 45th Annual Meeting of the Cognitive Science Society*.
- Danny Merckx and Stefan L. Frank. 2021. [Human sentence processing: recurrence or attention?](#) In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*.
- Lee Osterhout and Phillip J. Holcomb. 1992. [Event-related brain potentials elicited by syntactic anomaly](#). *Journal of Memory and Language*, 31(6):785–806.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [PyTorch: An imperative style, high-performance deep learning library](#). In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035.
- Stefanie Regel, Lars Meyer, and Thoomas C. Gunter. 2014. [Distinguishing neurocognitive processes reflected by P600 effects: Evidence from ERPs and neural oscillations](#). *PLoS ONE*, 9(5):e96840.
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. [Learning representations by back-propagating errors](#). *Nature*, 323(6088):533–536.
- Roland Schäfer. 2015. [Processing and querying large web corpora with the COW14 architecture](#). In P. Bański, H. Biber, E. Breiteneder, M. Kupietz, H. Lungen, and A. Witt, editors, *Proceedings of the 3rd Workshop on the Challenges in the Management of Large Corpora*, pages 28–34. Institut für Deutsche Sprache, Mannheim, Germany.
- Martin Schrimpf, Idan A. Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2021. [The neural architecture of language: Integrative modeling converges on predictive processing](#). *Proceedings of the National Academy of Sciences*, 118:e2105646118.
- Nathaniel J. Smith and Marta Kutas. 2015. [Regression-based estimation of ERP waveforms: I. The rERP framework](#). *Psychophysiology*, 52:157–168.
- Chara Tsoukala, Mirjam Broersma, Antal Van Den Bosch, and Stefan L. Frank. 2021. [Simulating code-switching using a neural network model of bilingual sentence production](#). *Computational Brain & Behavior*, 4:87–100.
- Stephan Verwijmeren, Stefan L. Frank, Hartmut Fitz, and Yung Han Khoe. 2023. [A neural network simulation of event-related potentials in response to syntactic violations in second-language learning](#). In *Proceedings of the 21st International Conference on Cognitive Modelling*.

On the communicative utility of code-switching

Yanting Li, Gregory Scontras, and Richard Futrell

Department of Language Science

University of California, Irvine

{yantil5, g.scontras, rfutrell}@uci.edu

Abstract

In the multilingual world we live in, code-switching (CS) is becoming more natural and more common. Why do bilingual language users CS from one language (the source language) to another (the target language) during communication, and how do they decide the CS point? In this corpus study, we investigate the hypothesis that it is harder to accurately express the meaning represented by the CS words in the source language. We analyzed sentences containing CS from Chinese–English bilingual corpora and found evidence for our hypothesis: compared to non-CS words, the English CS words are farther away from their closest Chinese word neighbors in a bilingual meaning space. This result supports the idea that bilinguals are using CS as a communication strategy to express their intended meanings accurately and efficiently.

1 Introduction

Code-switching (CS) refers to the scenario where a language user switches from one language to another during communication (Solorio et al., 2014; Adel et al., 2015; Zhou et al., 2020; Beatty-Martínez et al., 2020; Tomić and Valdés Kroff, 2022). The phenomenon is widely observed, both in spoken (e.g. Fricke and Kootstra, 2016; Heredia and Altarriba, 2001; Deuchar et al., 2014; Nguyen and Bryant, 2020) and written (e.g. Calvillo et al., 2020; Chang and Lin, 2014; Feldman et al., 2021; Chakravarthi et al., 2020) language use. Globalization has built stronger connections between countries and cultures; for English alone, there are over 1 billion people speaking it as a second language. The increase in multilingual speakers, together with the global status of English, has made CS involving English more and more common (Nakayama et al., 2018; Chakravarthi et al., 2020). As language scientists, we are charged with looking deeper into the process behind CS to better understand the communicative strategy of multilingual speakers.

Why do people code-switch? More specifically, what factors influence the choice to switch at certain words of an utterance but not others? Previous research has approached this question from different angles. Several factors have been shown to play a role in determining the CS point. For instance, word length: the longer a word, the more likely you are to switch to another language (where it may be shorter) to express that meaning (Myslín and Levy, 2015; Calvillo et al., 2020; Bhattacharya and van Schijndel, 2023). The syntactic role of the word is another factor: nouns are more likely to be CSed than verbs, function words, etc. (Myslín and Levy, 2015; Calvillo et al., 2020; Bhattacharya and van Schijndel, 2023). Semantic factors such as concreteness also play a role: more concrete words are more likely to be CSed (Myslín and Levy, 2015).

Another widely-discussed factor is predictability as operationalized by surprisal, the negative log probability of a word given context (Hale, 2001; Levy, 2008; Hale, 2016): CS words tend to have higher surprisal, meaning that these words are relatively less predictable from the context (Myslín and Levy, 2015; Calvillo et al., 2020; Bhattacharya and van Schijndel, 2023). There are two potential explanations for the role of surprisal in CS: according to a speaker-oriented explanation, words with higher surprisal impose more difficulty for production, and since speakers have limited cognitive resource, this will result in a weaker inhibition on the target language, letting words from that language “slip out” (Calvillo et al., 2020). Meanwhile, Myslín and Levy (2015) proposed an audience-oriented explanation: the words with higher surprisal need more attention from the listener, so the speaker will switch to a less frequent, and thus more salient language to alert the listeners of upcoming information peaks.

In this paper, we explore another aspect of efficiency: the communicative utility of CS words. Intuitively, the CS word in the language we switch

into might better express our intended meaning, as the source language may not have a word that expresses exactly the same meaning, even when there is a direct translation. For instance, 地下室 *dìxiàshì* in Mandarin Chinese is officially equivalent to English *basement*. However, the housing situation in China is very different from that of North America—there are far more tall apartment buildings than single-family homes in China. Because of this, when a Chinese–English bilingual hears the English word *basement*, the picture they have in mind might be different from the picture triggered by the Chinese word 地下室 *dìxiàshì*. Therefore in a Chinese conversation among Chinese–English bilinguals, when talking about the basement of a single-family home in the US, the speaker might consider switching into English for this word to achieve greater accuracy. In contrast, the English word *cat* expresses nearly exactly the same meaning as the Chinese word 猫 *māo* and so bilingual speakers may be less likely to CS for such a word. Similarly, Heredia and Altarriba (2001) provided an example in Spanish-English bilingual communication: the Spanish word *cariño* implies a combination of liking and affection, which cannot be expressed by an English word alone. Therefore, if a Spanish-English bilingual wish to refer to this concept, they would consider using Spanish to achieve a greater level of understanding.

In this research, we aim to test this hypothesis: people code switch when it is harder to express their intended meaning accurately in the source language.

2 Method

To see if a language has a vocabulary item that allows its speakers to express a certain meaning, we rely on word vectors, which help us locate words in a meaning space (Mikolov et al., 2013b,a; Bojanowski et al., 2017). In the meaning space, words with similar or related meanings are located close to each other while words with distant or unrelated meanings are located far away from each other. If there is a *bilingual* meaning space where words in both English and Chinese can be found, then for our hypothesis to be true, the English CS words should be located far away from any Chinese words in such a space, meaning no Chinese word has a meaning close enough to the CS words. To turn our hypothesis into something measurable, we choose to look for the closest Chinese word neighbor of

each CS word and calculate the 1) distance and 2) cosine similarity between the two. We will then do the same for the English translation of comparable non-CS words. We predict that, compared to non-CS words, the CS words have 1) longer distance to and 2) smaller cosine similarity with their closest Chinese word neighbor.

2.1 Materials

In order to conduct the above comparison, we need a bilingual meaning space for English and Chinese words. We also need a number of CS and non-CS words from natural language production.

Bilingual meaning space We use aligned word vectors to create the bilingual meaning space. While word vectors of a specific language can be used to locate words in the meaning space of that language, *aligned* word vectors are pre-trained to align meaning spaces of multiple languages (Smith et al., 2017; Conneau et al., 2018), so words from these languages can exist in the same space. We used the aligned word vectors of Chinese and English created by Bojanowski et al. (2017) and Joulin et al. (2018) based on the pre-trained vectors computed on Wikipedia. As the aligned word vectors are sorted by frequency, the top 150k English vectors and the top 150k Chinese ones are taken out and combined to create a bilingual vector space with 300k words. For any two word vectors, Chinese or English, in this space, their distance and cosine similarity tell us about how similar their meanings are to each other.

Code-switching corpora Two Chinese–English bilingual corpora are used: one written corpus and one spoken corpus. The written one consists of posts on Chinese international student forums of three universities in Pittsburgh (Calvillo et al., 2020). The content is mainly about housing, schooling, and life in Pittsburgh. The spoken corpus, on the other hand, is built on spontaneous multi-turn conversational dialogue sources collected in Hong Kong (Lovenia et al., 2022), covering topics on education, persona, philosophy, sports, and technology. In both corpora, native speakers of Mandarin Chinese (who also happen to be bilingual speakers of English) are communicating with each other, yet they choose to CS into their second language, English, at certain points.

In the written corpus, each CS sentence is paired with a structurally similar monolingual Chinese sentence. For instance:

CS sentence:

客厅还有一个小的balcony。

The living room also has a small balcony.

Matching sentence:

厨房面积大，还有一个小的吧台。

The kitchen size is big, and also has a small bar.

The two sentences have at least a 40% Levenshtein similarity of their POS sequences, and the matching sentence contains the same POS trigram as the CS point and the words before and after it (Calvillo et al., 2020). In this example, the word *balcony* and 吧台 *bātái* “bar” have the same POS tag and appear in a similar syntactic environment, but one is CSed while the other one is not, allowing us to make a close comparison of the word pair. Following the above two criteria, we found matching sentences for all CS sentences in the spoken corpus as well. If none of the monolingual Chinese sentences fulfilled both criteria for a CS sentence, the sentence was excluded from the analysis.

2.2 Procedure

We make the simplifying assumption that the words used in the actual language production, whichever language they are in, best express the intended meaning of the speaker. Based on this assumption, we extracted three groups of words from the corpora:

CS nouns While some instances of CS involve short phrases or compound words, we limited our focus to single-worded instances, specifically nouns. This is because only single words can be found in the bilingual meaning space, and nouns are the most likely to get CSed (Myslín and Levy, 2015; Calvillo et al., 2020; Bhattacharya and van Schijndel, 2023). We found 199 CS nouns in the written corpus and 531 in the spoken corpus that can be located within our bilingual meaning space.

English translations of matching non-CS nouns

As previously shown, each CS sentence in the corpora is paired with a syntactically similar monolingual Chinese sentence. This is to say, each CS noun (e.g., *balcony*) has a matching noun in the monolingual Chinese sentence (e.g., 吧台 *bātái* “bar”). We used googletrans (Han, 2020) to translate all these matching non-CS nouns into English. If a CS noun appears multiple times in the corpus, resulting in multiple matching non-CS nouns, we kept all that have a single-worded English translation that can be found in the bilingual meaning

space. If none of the matching words of a CS noun has a single-worded English translation, or none of the translations can be found in the meaning space, the CS noun was excluded. Take the word *basement* as an example: it appeared as a CS word in 8 different CS sentences in the written corpus, each matched with a different monolingual Chinese sentence. Therefore, there are 8 different matching non-CS nouns, namely 车 “car”, 客厅 “living room”, 里面 “inside”, 存储 “storage”, 屋内 “indoor”, 门口 “entrance”, 兼职 “part time” and 学校 “school”. Among these 8 non-CS nouns, only 6 have a single-worded English translation, and all 6 can be found in the bilingual meaning space, so these 6 words are kept as the matching nouns for *basement*. Meanwhile, for the word *balcony*, since it only appeared once in the whole corpus, it only has one match, which is *bar*. We ended up with 176 CS nouns in the written corpus and 477 in the spoken corpus with at least one matching non-CS noun.

English translations of random non-CS words

To create a larger pool of non-CS words that are not limited to nouns, we gathered all words that appear in the monolingual Chinese sentences from each corpus and kept the ones with single-worded English translations (according to googletrans; Han, 2020) that can be found in the bilingual meaning space. 1425 non-CS words remained for the written corpus and 2181 for the spoken corpus.

For each English word in the above three groups, we located the word in the bilingual meaning space and found the word in simplified Chinese located closest to it. We then used the vectors of both words to calculate their Euclidean distance as well as cosine similarity.

3 Analysis

CS nouns vs. non-CS nouns We conducted paired *t*-tests between the CS vs. non-CS noun pairs (e.g., *balcony* and *bar* in the example earlier). As some CS nouns appear multiple times in one corpus (e.g., *basement*), resulting in multiple matching non-CS nouns, five samples were randomly selected for a paired *t*-test. The CS nouns are the same across the samples, while the matching nouns may be different. This is to say, for *basement*, its matching noun could be *school* in sample 1, *storage* in sample 2, *car* for sample 3, etc. For both corpora, between the CS nouns and

Corpus	Sample	Distance	t Statistic	p -value	Cosine Similarity	t Statistic	p -value
written	CS	1.062	—	—	0.434	—	—
	non-CS 1	1.030	4.404	<0.001	0.468	-4.422	<0.001
	non-CS 2	1.030	4.335	<0.001	0.468	-4.368	<0.001
	non-CS 3	1.028	4.698	<0.001	0.469	-4.526	<0.001
	non-CS 4	1.029	4.559	<0.001	0.468	-4.556	<0.001
	non-CS 5	1.030	4.443	<0.001	0.468	-4.454	<0.001
spoken	CS	1.048	—	—	0.448	—	—
	non-CS 1	1.036	2.739	0.006	0.461	-2.821	0.005
	non-CS 2	1.036	2.886	0.004	0.461	-2.930	0.004
	non-CS 3	1.038	2.323	0.021	0.459	-2.349	0.019
	non-CS 4	1.034	3.179	0.002	0.463	-3.196	0.001
	non-CS 5	1.038	2.235	0.026	0.459	-2.269	0.024

Table 1: Mean distances and cosine similarities from English words to their nearest equivalents in Chinese. We show statistics from paired t -tests, comparing the actually-produced CS nouns against the non-CS nouns, for both measures. The labels of non-CS 1 through 5 represent the five samples of matching non-CS nouns that are randomly drawn. The $df = 476$ for the spoken corpus and $df = 175$ for the written corpus.

their closest Chinese word neighbors, the mean distance is significantly larger than that of non-CS nouns; the mean cosine similarity is significantly smaller (Table 1).

CS nouns vs. non-CS words In addition to the paired comparison between CS and matching non-CS nouns, we are also curious about whether CS nouns are different from non-CS words in general. Therefore, we used the boot library in R (Canty and Ripley, 2022; Davison and Hinkley, 1997) to bootstrap the 95% confidence interval of the mean distance and mean cosine similarity using the data of the English translations of non-CS words from both corpora ($n = 1425$ for the written corpus and $n = 2181$ for the spoken one). We then calculated the mean values of the CS nouns from each corpus ($n = 199$ for the written corpus and $n = 531$ for the spoken one) and examined whether they fall outside of the confidence intervals. The results are shown in Table 2 and visualized in Fig. 1. As we can see, the mean values of the CS nouns (the red dots in Fig. 1) are all outside of their corresponding 95% confidence interval.

4 Discussion

In this paper, we aimed to investigate why bilingual language users code switch during natural communication. We proposed that it is because of the communicative utility of CS and hypothesized that people choose to switch when it is harder to express their intended meaning accurately in the source language—there may not be a salient word in the source language that means the same as the CS word. While this may be a clear intuition for many bilingual speakers, we are not aware of any existing studies that measure this using naturalistic language production data. Here we proposed a way to quantitatively measure the communicative utility of CS. We tested our hypothesis by locating words from both languages in the same meaning space; the CS words in the target language should be far away from any words in the source language. Conversely, the cosine similarity between the CS word and its closest word neighbor in the source language should be small.

Our comparisons between the CS nouns vs. matching non-CS nouns and between the CS nouns vs. non-CS words in general show evidence

Corpus	Measure	Mean of CS nouns	95% confidence interval of non-CS words
written	Distance	1.064	(1.036, 1.043)
	Cosine Similarity	0.432	(0.454, 0.461)
spoken	Distance	1.047	(1.040, 1.045)
	Cosine Similarity	0.450	(0.452, 0.457)

Table 2: Mean distance and cosine similarity of CS nouns to their closest Chinese word neighbor in comparison to the bootstrapped 95% confidence interval of non-CS words. The data is visualized in Fig 1.

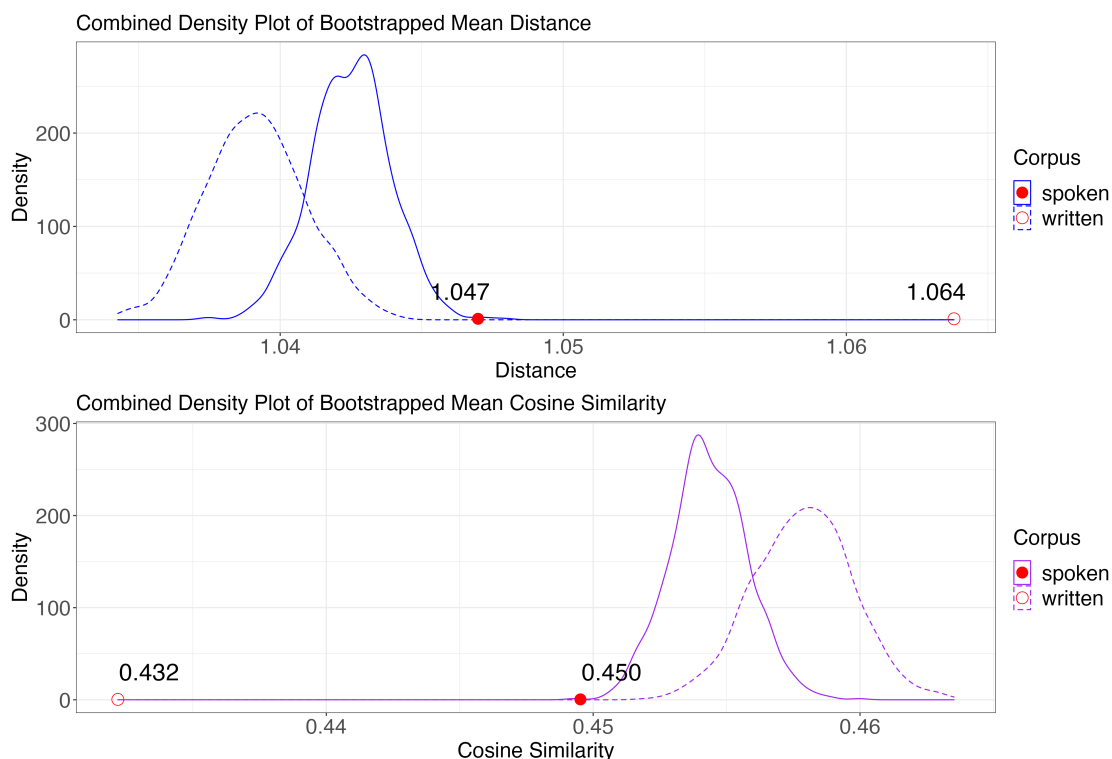


Figure 1: Density plots of the mean distance (top) and mean cosine similarity (bottom) bootstrapped from the non-CS words from the two corpora. The dashed line indicates data from the written corpus ($n = 1425$); the solid line indicates data from the spoken corpus ($n = 2181$). The red dots represent the mean values of the CS nouns, with the hollow dots for the written corpus ($n = 199$) and solid dots for the spoken corpus ($n = 531$).

supporting our hypothesis. Between the CS nouns in English and their closest Chinese neighbors, the distance is significantly larger and the cosine similarity is significantly smaller. This suggests that it is harder to pick a Chinese word to express the exact meaning of the English CS word. This is not to say that the meaning cannot be expressed accurately using Chinese at all—it might be possible if the speaker uses a combination of multiple Chinese words. However, CS is perhaps a faster, shorter, and therefore more efficient choice to achieve the communication goal.

It is worth noticing that the difference between the CS-nouns and non-CS nouns or words are consistently smaller for the spoken corpus when compared to those for the written corpus. One potential explanation for this trend is that people are under more time pressure when having a real-time spoken conversation compared to writing forum posts. This pressure means that when an English word expresses the intended meaning most accurately, even when there are Chinese words nearby in the meaning space, the speaker may not have enough

time to search for such words. As a result, they are more likely to produce CS. This is consistent with what was proposed by [Calvillo et al. \(2020\)](#), i.e. spoken language production allows CS to happen more frequently, although they see it as a result from the decreased cognitive resources to inhibit the alternative language. Another factor making CS more likely in spoken as opposed to written communication is that the switch cost is likely to be higher when typing than speaking, as it usually involves a switch of input keyboard. This cost will potentially create more resistance against CS, so typers are more motivated to search carefully in the meaning space around the English CS word for a Chinese equivalent, and only switch when it is sufficiently difficult to find anything with a close-enough meaning. These two factors, namely time pressure and switching cost, work in the same direction towards the difference we observed between the two corpora. This suggests that the mode of communication could affect the weight we assign to the communicative utility when making CS decisions.

Despite the above difference in effect size, the results from both corpora show consistent results that support our hypothesis. We thus contribute to the existing literature by identifying one more factor—the difficulty to accurately express a certain meaning in the source language—that may influence people’s decision on whether or not to CS, as well as where to switch. With CS becoming more popular all over the world, we hope to better explain this phenomenon and better understand CS as a communicative strategy that bilinguals utilize to achieve communication goals effectively and efficiently.

Acknowledgements

We thank Debasmita Bhattacharya and Marten van Schijndel for helpful discussion. We also thank the audience of the 2024 California Meeting on Psycholinguistics for their comments and feedback.

References

- Heike Adel, Ngoc Thang Vu, Katrin Kirchhoff, Dominic Telaar, and Tanja Schultz. 2015. [Syntactic and semantic features for code-switching factored language models](#). *IEEE/ACM transactions on audio, speech, and language Processing*, 23(1):431–440.
- Anne L Beatty-Martínez, Christian A Navarro-Torres, and Paola E Dussias. 2020. Codeswitching: A bilingual toolkit for opportunistic speech planning. *Frontiers in Psychology*, 11:1699.
- Debasmita Bhattacharya and Marten van Schijndel. 2023. Code-switching in online posts reveals information-theoretic audience design. In *Human Sentence Processing 2023*, Pittsburgh, PA.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Jesús Calvillo, Le Fang, Jeremy Cole, and David Reitter. 2020. [Surprisal predicts code-switching in chinese-english bilingual text](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4029–4039.
- Angelo Canty and B. D. Ripley. 2022. *boot: Bootstrap R (S-Plus) Functions*. R package version 1.3-28.1.
- Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadarshini, and John P McCrae. 2020. Corpus creation for sentiment analysis in code-mixed tamil-english text. *arXiv preprint arXiv:2006.00206*.
- Joseph Chee Chang and Chu-Cheng Lin. 2014. [Recurrent-neural-network for language detection on twitter code-switching corpus](#). *CoRR*, abs/1412.4314.
- Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. [Word translation without parallel data](#).
- A. C. Davison and D. V. Hinkley. 1997. *Bootstrap Methods and Their Applications*. Cambridge University Press, Cambridge. ISBN 0-521-57391-2.
- Margaret Deuchar, Peredur Davies, Jon Herring, M Carmen Parafita Couto, and Diana Carter. 2014. Building bilingual corpora. *Advances in the Study of Bilingualism*, pages 93–111.
- Laurie Beth Feldman, Vidhushini Srinivasan, Rachel B. Fernandes, and Samira Shaikh. 2021. [Insights into codeswitching from online communication: Effects of language preference and conditions arising from vocabulary richness](#). *Bilingualism: Language and Cognition*, 24(4):791–797.
- Melinda Fricke and Gerrit Jan Kootstra. 2016. [Primed codeswitching in spontaneous bilingual dialogue](#). *Journal of Memory and Language*, 91:181–201.
- John Hale. 2001. [A probabilistic earley parser as a psycholinguistic model](#). In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1–8. Association for Computational Linguistics.
- John Hale. 2016. [Information-theoretical complexity metrics](#). *Language and Linguistics Compass*, 10(9):397–412.
- SuHun Han. 2020. [googletrans 3.0.0](#).
- Roberto R Heredia and Jeanette Altarriba. 2001. Bilingual language mixing: Why do bilinguals code-switch? *Current Directions in Psychological Science*, 10(5):164–168.
- Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Roger Levy. 2008. [Expectation-based syntactic comprehension](#). *Cognition*, 106(3):1126–1177.
- Holy Lovenia, Samuel Cahyawijaya, Genta Indra Winata, Peng Xu, Xu Yan, Zihan Liu, Rita Frieske, Tiezheng Yu, Wenliang Dai, Elham J Barezi, et al. 2022. [Ascend: A spontaneous chinese-english dataset for code-switching in multi-turn conversation](#). In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. [Efficient estimation of word representations in vector space](#).
- Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. [Distributed representations of words and phrases and their compositionality](#). *CoRR*, abs/1310.4546.

- Mark Myslín and Roger Levy. 2015. [Code-switching and predictability of meaning in discourse](#). *Language*, pages 871–905.
- Sahoko Nakayama, Takatomo Kano, Quoc Truong Do, Sakriani Sakti, and Satoshi Nakamura. 2018. [Japanese-english code-switching speech data construction](#). In *2018 Oriental COCOSDA - International Conference on Speech Database and Assessments*, pages 67–71.
- Li Nguyen and Christopher Bryant. 2020. [CanVEC - the canberra Vietnamese-English code-switching natural speech corpus](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4121–4129, Marseille, France. European Language Resources Association.
- Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. [Offline bilingual word vectors, orthogonal transformations and the inverted softmax](#).
- Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. [Overview for the first shared task on language identification in code-switched data](#). In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.
- Aleksandra Tomić and Jorge R. Valdés Kroff. 2022. [Expecting the unexpected: Codeswitching as a facilitatory cue in online sentence processing](#). *Bilingualism: Language and Cognition*, 25(1):81–92.
- Xuehao Zhou, Xiaohai Tian, Grandee Lee, Rohan Kumar Das, and Haizhou Li. 2020. [End-to-end code-switching tts with cross-lingual language model](#). In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7614–7618.