Preface: SCiL 2022 Editors' Note

Allyson Ettinger[1], Tim Hunter[2], and Brandon Prickett[3]
[1]University of Chicago, [2]University of California, Los Angeles, [3]UMass Amherst

This volume contains research presented at the fifth annual meeting of the Society for Computation in Linguistics (SCiL), held online, February 7-9, 2022.

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 38 submissions to the conference, 15 abstracts and 23 papers. 18 submissions were selected for oral presentation (~47%) and 8 for poster presentation (~21%).

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

> Jonathan Brennan, Robert Frank, Jeffrey Heinz, William Idsardi, Tamar Johnson, Itamar Kastner, Maayan Keshev, Dongsung Kim, Jordan Kodner, Nur Lan, Terry Langendoen, Ling Liu, Yang Janet Liu, Giorgio Magri, Timothee Mickus, Yohei Oseki, Tiago Pimentel, Ollie Sayeed, Carolyn Anderson, Shohini Bhattasali, Canaan Breiss, Alicia Chatten, Xiaoli Chen, Alexander Clark, Lindy Comstock, Aniello De Santo, Brian Dillon, Matt Goldrick, Kyle Gorman, Thomas Graf, Bruce Hayes, Kasia Hitczenko, Nick Huang, Adam Jardine, Tracy Holloway King, Christo Kirov, Andrew Lamont, Tal Linzen, Robert Malouf, Connor Mayer, R. Thomas Mccoy, Emily Morgan, Max Nelson, Charlie O'Hara, Lisa Pearl, Laurel Perkins, Christopher Potts, Nathan Schneider, Sebastian Schuster, Caitlin Smith, Alex Warstadt, Bonnie Webber, Colin Wilson, Qihui Xu, Olga Zamaraeva, Yilun Zhu, Katrin Erk, Lucy Li, Alicia Parrish, Edward Stabler, Aaron Steven White, and Adina Williams.

Thanks also to Joe Pater and Erin Jerome for logistical help.

SCiL 2022 also included invited talks by Alexander Clark (King's College London), Katrin Erk (University of Texas at Austin), and Caitlin Smith (University of California, Davis). Further information can be found at our website: https://blogs.umass.edu/scil/.

A Model Theoretic Perspective on Phonological Feature Systems

Scott Nelson
Stony Brook University
Department of Linguistics
Institute for Advanced Computational Science
scott.nelson@stonybrook.edu

Abstract

This paper uses model theory to analyze the formal properties of three phonological feature systems: privative, full binary, and underspecified. By systematically manipulating the choice of logical language and representational primitives, it is shown that logical negation effectively converts any feature system into a full binary one. This further implies that in order to have underspecification or non-binary feature oppositions, valuation should be encoded into the representational primitives rather than derived through the logical connectives. These results are obtained by comparing the predicted natural classes of each formalization.

1 Introduction

Phonological features are present in some form in most modern theories of phonology. While there are debates over how to best represent features, it is typically agreed that features encode sub-segmental acoustic and/or articulatory properties. A feature system is a set of valued features where the valuation is typically drawn from the set $\{+, -\}$. Segments are therefore shorthand for collections of valued features, and rules or constraints use features to target groups of sounds that undergo the same phonological processes.

In practice, feature systems also regularly contain a 0 valuation to imply that a certain segment is not specified as either $+$ or $-$ for a given feature. The 0 valuation seems to serve two theoretical purposes. The first purpose is simply as a placeholder for a feature that does not apply for a given segment. For example, the feature [distributed] differentiates between coronal segments made with the tip or blade of the tongue. For non-coronal sounds, this distinction is meaningless and therefore is usually represented with 0. The second purpose that 0 serves is for underspecification. Feature systems that use underspecification do so in order to ensure that certain rules do not target specific segments, even though they share a certain phonetic property. For example, sonorant sounds are phonetically voiced, but in some feature systems they are phonologically underspecified as [0 voice] which allows them to be excluded from phonological voicing assimilation processes.

Most feature systems mix $\{+, -, 0\}$ in different ways, but it is not clear whether or not each system can be formally represented and interpreted in the same way. It is also worth considering whether or not each feature system is meaningfully different than the others, or if it can be thought of as a notational variant. One set of tools that allows for looking into these types of questions is model theory and logic. Model theory is a branch of mathematics that allows for the precise definition of relational structures such as strings (Libkin, 2004). These structures can be further evaluated using different types of logic.

Model theory and logic can therefore provide a meta-language to compare different types of phonological representations. Strother-Garcia (2019) compares different types of syllabic representations, Jardine et al. (2021) study the difference between traditional autosegmental representations (Goldsmith, 1976) and Q-theoretic representations (Inkelas and Shih, 2016), and Oakden (2020) shows how different representations of tone are essentially notational variants. Another advantage of using model theory for phonology is that it has a well defined relationship with computational complexity and learnability (Strother-Garcia et al., 2016; Vu et al., 2018; Chandlee et al., 2019). Additionally, it provides a way to formalize differences between representational structures so that we can move beyond relying solely on our intuitions. Phonological feature systems are one area that has yet to be explored in this way.

In this paper I will use model theory and logic

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 1-10.
Held on-line February 7-9, 2022

to explore three types of feature systems: a privative system that uses $\{+, 0\}$, a full binary system which uses $\{+, -\}$, and a contrastive system that uses $\{+, -, 0\}$. While Mayer and Daland (2020) provide different algorithms for a how these feature systems could be learned, this paper focuses on how each feature system can be formally represented using different types of logics and representational primitives. The diagnostic that I will focus on are the natural classes that are expected for a given feature system.

Previous mathematical treatments of feature systems have primarily focused on the binary aspect of features (Bale and Reiss, 2018; Keenan and Moss, 2009; Johnson, 1972). Their systems look like the full binary system where every segment is specified as either $+$ or $-$ for every feature. The reason for this is due to their use of logical negation. The main result I will show is that a full binary feature system is the only possible result when using logical negation. Consequently, in order to effectively have 0 values, the positive/negative feature valuation must be encoded into the representational primitives rather than emerge from the logical connectives.

2 Phonological Features

The use of phonological features as a tool for phonological analysis is typically traced back to the Prague School, notably Nikolai Trubetzkoy and Roman Jakobson. Trubetzkoy (1939) proposed three different types of feature based oppositions: privative, gradual, and equipollent. A privative feature in his analysis would be [voice] where a segment either has the property of being voiced or it does not. Gradual features are things such as [height 1], [height 2], ..., [height n] where the numerical valuations encode the vowel height scale. Equipollent features are similar to privative features in that they are present or absent, but unlike privative features they do not encode a binary-like opposition. The examples of features used to explain an equipollent opposition are [labial], [coronal], and [dorsal].

Jakobson's contribution to feature theory culminated with *Preliminaries to Speech Analysis* (Jakobson et al., 1951). In this monograph, all features were treated as binary, specifically encoded as being either $+$ and $-$ for each feature. The use of binary features in phonology was further amplified due to their inclusion in *The Sound Pattern of English* (Chomsky and Halle, 1968) and they continue to be used as the default valuation of features in most modern phonology textbooks.

As feature theory has developed over the last decade, there have been debates along multiple dimensions about how best to represent features. One dimension is whether or not features should be thought of as attributes or particles in the terms of Ladd (2014). That is, should we think about features in terms of feature bundles that are ordered temporally, or should we think about them in autosegmental terms where each feature is specified on its own tier and has some type of relation to a general timing unit. In this paper I will focus on the former as it is more typical. Nonetheless, the results of this study should be able to be generalized to autosegmental or feature geometric systems (McCarthy, 1988).

A second dimension in the debate on features has to do with whether or not features should be thought of as discrete or gradient categories. The gradient approach often is lumped in with a scalar approach (e.g., Flemming, 2001), but it is possible to have scalar features without forgoing discrete categories. Since I am using finite model theory in this paper, the feature set needs to be finite and therefore discrete categories are necessary. However, it is also possible to approximate gradient feature values by having a large, but finite, set of possible numerical valuations.

Two other debates have to do with whether or not features are innate or emergent (Mielke, 2008), or whether or not features contain phonetic substance or instead are substance free (**?**). Neither of these two areas directly affect feature valuation and will be left aside.

2.1 Natural Classes

Natural classes are the result of partitioning a language's segment inventory using phonological features. Traditionally, there are two explanations for natural classes. The phonetic explanation is that all segments that form a natural class share one or more phonetic property. The distributional explanation is that all segments that form a natural class are the target/trigger for a phonological process or involved in some type of constraint. One problem with these explanations is that they do not always cohere (Duanmu, 2016). For example, there are groups of segments that have the same distribution, but nonetheless do not share a phonetic property. Mielke (2008, p. 12) attempts to explain this disconnect by arguing for emergent features. He also offers the following definition that will be useful for current purposes: a natural class is, "a group of sounds in an inventory which share one or more distinctive features, *within a particular feature theory* to the exclusion of all other sounds in the inventory," (emphasis original).

It also should be clarified what it means for two segments to share a feature. As Bale and Reiss (2018) point out, phonologists can be sloppy when talking about features by not clearly distinguishing the difference between a feature and a segment's specific valuation for a feature. It is safe to assume that what Mielke means in the quote above is that two segments are part of the same natural class when they share the same *valuation* for one or more distinctive features. In set theoretic terms we can think of natural classes as groups of sounds whose valued feature intersection is non-empty.

Conjunction would therefore be the logical parallel to this and conjunction does seem to be the way in which phonologists tend to think about natural class formation. For example, Kenstowicz and Kisseberth (1979, p. 241) write, "...an adequate feature system should permit any natural class of sounds to be represented by the conjunction of features in a matrix," while Odden (2005, p. 49) writes, "Natural classes can be defined in terms of conjunctions of features...." If a phoneme /n/ had the feature bundle [+coronal +sonorant -continuant +nasal], then we say that /n/ has the properties of being +coronal AND + sonorant AND -continuant AND +nasal.

While conjunction is the main way in which features seem to be combined to form natural classes, there are other possible logical connectives that one might use. For example, the curly brackets {} were used by Chomsky and Halle (1968) to indicate disjunctive triggering environments. Furthermore, Mielke (2008) showed that ∼97% of the phonologically active classes he looked at can be described with the *SPE* feature system if disjunction is allowed. This is an increase of 26% from *SPE*'s coverage without disjunction, which seems like a positive finding, but if we abstract away from the specific features used in any given system, disjunction should be able to cover 100% of natural classes.[1] One reason why we may not want disjunction, despite its ability to allow for broad empirical coverage, is the fact that with arbitrary disjunction any subset of segments can form a natural class.

While logical negation can be interpreted as complementation, a reviewer points out that its use for defining natural classes has largely been avoided (e.g. Chomsky and Halle (1968)). A notable exception is Hayes and Wilson (2008) which employed a complementation operator in their definition of constraints.

Quantification is another tool used in formal logic that could be used for interpreting feature bundles. For the most part, phonologists seem to stay away from quantification, but Reiss (2003) uses it to define identity relations in the structural description of phonological rules. Since the structural description is usually thought to be a natural class, this could be one area where quantification is used for interpreting feature bundles. That being said, identity is often baked into the axioms of logical interpretation languages. Strother-Garcia (2019) discusses the relationship between quantifier-free logics and locality for syllabification, but it is worth pursuing whether or not this is the right approach when considering phonological features. This is left for future work.

2.2 Underspecification

0 values are often associated with underspecification. Underspecification is when certain features are not assigned either a plus or a minus value for a given feature. Two common types of underspecification are privative underspecification where minus values are completely eliminated and only + feature values are assigned, and contrastive underspecification where any feature value that is redundant is removed. For example, sonorants can have a 0 value for the [voice] feature since sonorants do not have a voicing contrast and are by default [+voice]. The redundant value for [voice] is then filled back in at the end of the derivation. Sonorants being underspecified for [voice] has been a central argument in the debate around contrastive underspecification and will be used in the current analysis as well (see Steriade (1995) for further discussion and review).

0 values can also be used for non-redundant purposes. This is sometimes used when a feature only applies to a certain class of sounds (Hayes, 2011). Steriade (1995, p. 117) calls this "trivial" underspecification, contrasting it with the "temporary" underspecification described in the previous para-

[1] The reason that *SPE* (and the other feature systems evaluated by Mielke (2008)) do not reach 100% is because they are unable to contrast between certain types of segments such as pre-nasalized/post-nasalized stops.

graph. In the analysis in this paper, the distinction between trivial and temporary underspecification collapses because only the natural classes the phonological feature matrices represent is under consideration.

3 Model Theory

3.1 String Models

Strings can be straightforwardly defined in model theory. At minimum, a model theoretic representation includes a finite domain $\mathcal{D}$ and a finite set of relations $\mathcal{R}$. $\mathcal{R}$ also typically includes a set of labeling relations drawn from a primitive set of symbols Σ onto elements of the domain, and an ordering relation used to structure the domain elements. Σ is typically referred to as the alphabet since it contains the segmental labels for the domain elements. I will use the successor ordering relation throughout this paper. The domain is typically taken to be the natural numbers $\mathbb{N}$. Given this, we can define successor as $\langle i, i+1 \rangle \in \mathcal{D} \times \mathcal{D}$. A model signature is a tuple containing all of this information. For the successor model $\mathcal{M}^{\lhd}$, this contains $\langle \mathcal{D}, \mathcal{R}_\sigma | \sigma \in \Sigma, \lhd \rangle$.

$$\begin{aligned} \mathcal{D} &= \{1,2\} \\ \mathcal{R}_a &= \{2\} \\ \mathcal{R}_b &= \{1\} \\ \lhd &= \{\langle 1,2 \rangle\} \end{aligned}$$

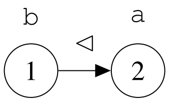


Figure 1: Successor word model for $\mathcal{M}^{\lhd}_{ba}$.

Figure 1 shows the successor word model for the word ba given the alphabet $\Sigma = \{a, b\}$. This defines the word over segments. As phonologists we may want to analyze this structure using features, but since features are not innate to the model, we have to define them ourselves. One way to do this is with user defined predicates. These are predicates that the analyst imposes on the model. Features can be defined disjunctively from a segment based model such as the one in Figure 1. For example, we define the predicate voi as:

(1) $\mathsf{voi}(x) \overset{\text{def}}{=} \mathcal{R}_a(x) \vee \mathcal{R}_b(x)$

This formula says that any segment that is labeled as a or b has the property of being voiced. Features are therefore epiphenomenal in this type of model.

A second option is to have our alphabet Σ be made of phonological features rather than phonological segments. This also requires a change to the labeling relations. Typically, each domain element is given a single label. If phonological features are the primitives, then it must be the case that a single domain element can have more than one label. Figure 2 shows a second successor model for the word ba, this time using features as the alphabetic primitives rather than segments. With this type of model we can define segments conjunctively using features. In this case, it is the segments which are epiphenomenal.

$$\begin{aligned} \mathcal{D} &= \{1,2\} \\ \mathcal{R}_{\texttt{+voi}} &= \{1,2\} \\ \mathcal{R}_{\texttt{+lab}} &= \{1\} \\ \mathcal{R}_{\texttt{-cont}} &= \{1\} \\ \mathcal{R}_{\texttt{-syl}} &= \{1\} \\ \mathcal{R}_{\texttt{+syl}} &= \{2\} \\ \mathcal{R}_{\texttt{+back}} &= \{2\} \\ \mathcal{R}_{\texttt{-high}} &= \{2\} \\ \lhd &= \{\langle 1,2 \rangle\} \end{aligned}$$

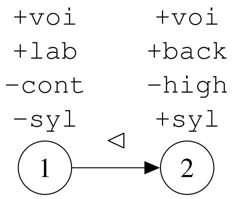


Figure 2: Successor with features model for *ba*.

In the example given here, *valued* features make up the primitive units. That is, $+$ and $-$ values are built directly into the individual labeling relations. Another possibility would be to only have feature labels as the set of primitives and interpret feature valuation as whether or not a domain element has a given feature label. For example, a [+voice] domain element would be one that is labeled with the feature [voice] while a [-voice] domain element would be one that does not have the label [voice]. I will refer to the first style, where the $+/-$ values are encoded directly into the primitives, as *bivalent primitives*. The second style, where only a feature itself is encoded into the primitives, will be referred to as *univalent primitives*.

3.2 Logical Evaluation

Model theoretic structures can be interpreted with different types of logic. First-order logic (FO) is commonly used, but it allows for quantification which does not seem to be used when describing phonological natural classes. Quantifier-free logic (QF) is like FO except without quantifiers. Even this is likely too powerful since it still uses standard logical connectives like conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), and implication ($\rightarrow$). Conjunction and possibly negation are the only primitives that seem to be required for defining natural classes and yet if they are both allowed to freely combine we can then derive the other logical connectives. For example, disjunction $(P \vee Q)$ can be defined as $\neg(\neg P \wedge \neg Q)$. One solution to this problem is to restrict the use of negation to only atomic sentences. We can do so by defining different types of sub-QF logics.

Of the three logics we can define this way, two of them will be used in this paper. Conjunction of negative and positive literals (CNPL) allows for the conjunction of any sentence within the language, but negation is only allowed to scope over atomic predicates. Conjunction of positive literals (CPL) only allows for the conjunction of sentences. Negation is strictly excluded from the logical language. Conjunction of negative literals (CNL) is the third logical language and only allows for negated atomic primitives to be combined with conjunction. The syntax of CNPL and CPL are recursively defined in (2).

(2) (a) CNPL
- i. Base case: For all atoms P, "P" and "$\neg P$" are sentences.
- ii. Inductive case: For all sentences A, B, "$A \wedge B$" is a sentence.

(b) CPL
- i. Base case: For all atoms, P, "P" is a sentence.
- ii. Inductive case: For all sentences A, B, "$A \wedge B$" is a sentence.

For this paper, the atoms are the feature labeling relations.

4 Model Theoretic Feature Systems

In this section I will demonstrate how different phonological feature systems can be expressed using model theory. The diagnostic used in this analysis is a comparison of the natural classes that a certain feature system is predicted to have based on a feature table versus what type of natural classes can be formed from the model theoretic representation and interpretation. Phonological feature systems are typically presented as tables of $+$, $-$, and 0 values. Three examples are shown in Table 1 (adapted from Mayer and Daland (2020)).

The privative feature system uses only $+$ and 0, the full binary system uses only $+$ and $-$, and the contrastive system uses $+$, $-$, and 0. Each of these therefore predicts different sets of natural classes. Since the 0 value typically represents the lack of a valuation, the natural classes for each feature system are based on similarity of $+$ and $-$ values. The set of natural classes for the privative feature system is therefore {{N},{N,D}}. There are in fact two ways to define the subset {N} in this feature system: segments that are [+son] or segments that are [+son, +voi]. The subset {N,D} is defined as all segments that are [+voi]. The set of natural classes for the full system is {{N}, {N,D}, {D}, {T}, {D,T}} and the set for the contrastive system is {{N}, {D}, {T}, {D,T}}. Construction of these sets was done the same way as described for the privative feature system.

	Privative		Full		Contrastive	
	son	voice	son	voice	son	voice
N	+	+	+	+	+	0
D	0	+	-	+	-	+
T	0	0	-	-	-	-

Table 1: Example of three types of feature systems. N represents all sonorants, D represents voiced obstruents, and T represents voiceless obstruents.

There are two reasonable ways in which we can turn these feature tables into model theoretic representations. The first way would be to use a segmental model and define translations from the segmental model into different feature models. MSO-definable string to string transformations (Courcelle, 1994; Engelfriet and Hoogeboom, 2001) allow for translation between different representational systems. A second way would be to use the feature successor model and have the difference in valuations emerge from the definitions of each specific model. Both methods will result in the same structures for evaluation, but I will take the second approach as it aligns more directly with the theme and discussion of the paper so far.

The primary model signature that will be used is the successor model defined above: $\mathcal{M}^{\lhd} = \langle \mathcal{D}, \mathcal{R}_{\sigma} | \sigma \in \Sigma, \lhd \rangle$. We can alter the general successor model slightly by providing fixed labeling relations. This allows for the definition of two model signatures: a univalent primitive signature $\mathcal{M}^{\upsilon}$ and a bivalent primitive signature $\mathcal{M}^{\beta}$. These are defined as follows:

(3) $\mathcal{M}^{\upsilon} = \langle \mathcal{D}, \texttt{voi}, \texttt{son}, \lhd \rangle$

(4) $\mathcal{M}^{\beta} = \langle \mathcal{D}, +\texttt{voi}, +\texttt{son}, -\texttt{voi}, -\texttt{son}, \lhd \rangle$

We can further specify models for each feature system (privative = P, full = F, contrastive = C).

This leaves us with six potential structures: $\mathcal{M}^{v}_{\text{P}}$, $\mathcal{M}^{v}_{\text{F}}$, $\mathcal{M}^{v}_{\text{C}}$, $\mathcal{M}^{\beta}_{\text{P}}$, $\mathcal{M}^{\beta}_{\text{F}}$, and $\mathcal{M}^{\beta}_{\text{C}}$. Each can then be evaluated using CPL and CNPL.

Since these models define strings, I will define the string DNT.[2] For the univalent primitive signature ($\mathcal{M}^{v}$), I will assume that any segment with a $+$ value in the feature table will be labeled with that feature. In this case, both 0 and $-$ values do not correspond to a label. For the bivalent primitive signature ($\mathcal{M}^{\beta}$), I will assume that any segment with a $+$ value in the feature table will be assigned the $+f$ label and any segment with a $-$ value in the feature table will be assigned the $-f$ label. The 0 once again will correspond to no label.

4.1 Evaluating Univalent Primitive Models

Let us start by looking at the different univalent feature models as interpreted with CPL logic. As it turns out, the privative and full features systems have an identical structure under $\mathcal{M}^{v}$. This is not all that surprising since a privative model just replaces all of the $-$ values with 0 values. In other words, both types of feature system allow for binary distinctions to be made, but the full feature system does it explicitly with a $-$ while the privative system does it through presence/absence of a feature. The top of Figure 3 shows the model for the string DNT.

As can be seen, domain element 1 which corresponds to D is only labeled with the `voi` label while domain element 2 which corresponds to N is labeled with both the `voi` and `son` labels. Domain element 3 is left unlabeled since T has no corresponding $+$ features in either the privative or full feature charts. The model for the contrastive feature system is shown in the bottom of Figure 3.[3] It differs slightly from the first model signature due to N having a 0 value for `voi` since voicing is not contrastive for sonorants in this feature system. Because of this, domain element 2 only receives the `son` label.

Given these model theoretic structures, we can now interpret them logically. Since our first evaluation logic is CPL, we can look at which domain elements satisfy all of the predicates we can make using conjunction over positive literals. The primitives are the features `voi` and `son`, so there are three predicates: the singletons `voi` and `son`, as well as the conjunction of the two `voi` $\wedge$ `son`. Table 2 shows the resulting classes of sounds from interpreting the structures in this way.

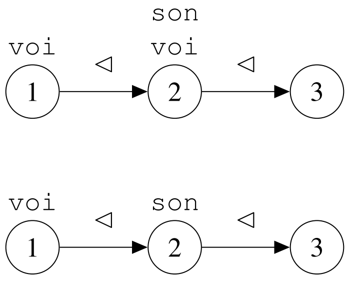


Figure 3: Models for the string DNT using models $\mathcal{M}^{v}_{\text{P}} = \mathcal{M}^{v}_{\text{F}}$ (top) and $\mathcal{M}^{v}_{\text{C}}$ (bottom)

CPL($\mathcal{M}^{v}$)	$\mathcal{M}^{v}_{\text{P}}$	$\mathcal{M}^{v}_{\text{F}}$	$\mathcal{M}^{v}_{\text{C}}$
`voi`	{N,D}	{N,D}	{D}
`son`	{N}	{N}	{N}
`son` $\wedge$ `voi`	{N}	{N}	{}
MISSING	–	{D}, {T}, {D,T}	{T}, {D,T}
EXTRA	–	–	–

Table 2: CPL logical interpretation of the different univalent primitive model theoretic structures.

The classes for $\mathcal{M}^{v}_{\text{P}}$ and $\mathcal{M}^{v}_{\text{F}}$ are {{N},{N,D}}. For $\mathcal{M}^{v}_{\text{P}}$, which is the correlate of the privative feature system, this is the expected result. That is, it matches the set of natural classes that we would predict given the feature table in Table 1. For $\mathcal{M}^{v}_{\text{F}}$, this is an under-prediction. As can be seen in the MISSING row of Table 2, $\mathcal{M}^{v}_{\text{F}}$ as interpreted with CPL fails to generate the classes {{D},{T},{D,T}} which a full binary feature system should have. The reason these classes are not generated is because they require an ability to reference minus values in some way. This is not possible given the CPL with univalent primitive system used here. $\mathcal{M}^{v}_{\text{C}}$ as interpreted with CPL correctly rules out the class {D,N}, which is one of the primary goals of the contrastive feature system, but still under predicts in a similar way to the full model. This once again has to do with not being able to reference minus values for natural class formation.

Overall, CPL logic with univalent primitives is a good way of representing privative feature systems since the lack of minus features aligns with CPL's inability to target minus features. For the other two feature systems, we need to be able to reference minus features in order to obtain the desired natural classes. One way that this may be accomplished

[2] Since N indicates all sonorant sounds this could correspond to words like *bus* or *juice*.

[3] I will only show the visual representation of models in the main body of the paper from here on out.

is through the use of negation. As mentioned in a previous section, we want to limit our negation to the atomic elements, which in this case are feature values. This allows for a straightforward interpretation such that atomic elements on their own can be thought of as $+F$ for some atomic feature and negated atomic elements can be thought of $-F$ for the same feature. CNPL as our interpretation logic allows us to take this route.

CNPL($\mathcal{M}^v$)	$\mathcal{M}^v_{\text{P}}$	$\mathcal{M}^v_{\text{F}}$	$\mathcal{M}^v_{\text{C}}$
`voi`	{N,D}	{N,D}	{D}
$\neg$`voi`	{T}	{T}	{N,T}
`son`	{N}	{N}	{N}
$\neg$`son`	{D,T}	{D,T}	{D,T}
`son` $\wedge$ $\neg$`son`	{}	{}	{}
`son` $\wedge$ `voi`	{N}	{N}	{}
`son` $\wedge$ $\neg$`voi`	{}	{}	{N}
$\neg$`son` $\wedge$ `voi`	{D}	{D}	{D}
$\neg$`son` $\wedge$ $\neg$`voi`	{T}	{T}	{T}
`voi` $\wedge$ $\neg$`voi`	{}	{}	{}
MISSING	–	–	–
EXTRA	{D}, {T}, {D,T}	–	{N,T}

Table 3: CNPL logical interpretation of the different univalent primitive model theoretic structures.

Table 3 shows the interpretation of the $\mathcal{M}^v$ structures using CNPL logic. Once again the privative and full feature system models will have the same set of classes: {{N}, {N,D}, {D}, {T}, {D,T}}. In this case, this is the set of classes that we would expect for the full feature system. This means that the privative model now overpredicts in regards to natural class formation. As can be seen in the EXTRA row of Table 3, the classes {{D}, {T}, {D,T}} are generated because the use of negation effectively turns every feature into a binary feature. For the contrastive feature system, this also presents a problem. In the contrastive system, a distinction needs to be made between the negative value for a feature and the lack of any value for a feature. Logical negation collapses this distinction. As can be seen in the third column, $\mathcal{M}^v_C$ considers N to be part of the $\neg$`voi` class. So not only does CNPL with univalent features over predict in the case of the contrastive feature system model, it over predicts by creating a class that none of the three feature systems uses.

A univalent model interpreted with CNPL therefore best models a full feature system where every segment is fully specified for either + or −. Since the privative and full feature systems have the same model signature in this analysis, the meaningful difference between these two systems seems to be in how the structures are interpreted logically rather than how the structures are labeled.[4] It also appears that there is no way to accurately represent a contrastive feature system with univalent primitives using either of the two interpretation logics. For contrastive feature systems it is necessary to target minus feature values when defining natural classes, but it is also necessary to maintain the distinction between a 0 value and a − value. One way in which this may be accomplished is to strictly encode the feature valuation into the primitives rather than using logical negation to explain the +/− distinction.

4.2 Evaluating Bivalent Primitive Models

Figure 4 shows the models for $\mathcal{M}^{\beta}_{\text{P}}$, $\mathcal{M}^{\beta}_{\text{F}}$, and $\mathcal{M}^{\beta}_{\text{C}}$. Recall that in $\mathcal{M}^{\beta}$, the primitives include $+$`voi`, $+$`son`, $-$`voi`, and $-$`son`. Each of the three model signatures varies in how much information is encoded. For all models, a $+$ value for a feature results in a label of $+F$ and a $-$ value for a feature results in a label of $-F$. Unlike the univalent models, each feature system here does result in a unique model theoretic structure. This means that the difference between the feature systems cannot be explained by the logical interpretation language.

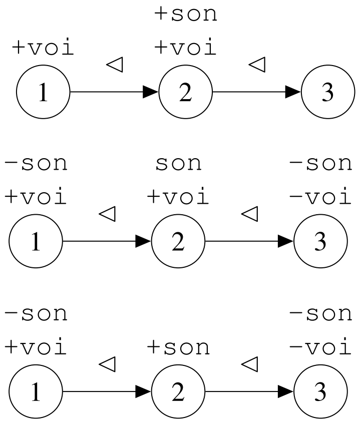


Figure 4: Models for the string DNT using models $\mathcal{M}^{\beta}_{\text{P}}$ (top), $\mathcal{M}^{\beta}_{\text{F}}$ (middle), and $\mathcal{M}^{\beta}_{\text{C}}$ (bottom).

Given these structures, we can once again interpret them logically using CPL. Table 4 shows what

[4] If we were to use negative valued feature labels as our univalent primitives this distinction may not hold. In this case, all segments would be unlabeled for the privative feature system. So it is not necessarily any univalent feature model where the distinction between privative and full feature systems is in the logical interpretation, but rather a univalent feature model that encodes positive feature values.

classes result from the models. Notice that each model does not contain any extra, nor have any missing, classes. As it turns out, the interpretation of each model now results in the exact set of natural classes that the corresponding feature set predicts.

CPL($\mathcal{M}^\beta$)	$\mathcal{M}^\beta_{\text{P}}$	$\mathcal{M}^\beta_{\text{F}}$	$\mathcal{M}^\beta_{\text{C}}$
+voi	{N,D}	{N,D}	{D}
−voi	{}	{T}	{T}
+son	{N}	{N}	{N}
−son	{}	{D,T}	{D,T}
+son ∧ −son	{}	{}	{}
+son ∧ +voi	{N}	{N}	{}
+son ∧ −voi	{}	{}	{}
−son ∧ +voi	{}	{D}	{D}
−son ∧ −voi	{}	{T}	{T}
+voi ∧ −voi	{}	{}	{}
MISSING	–	–	–
EXTRA	–	–	–

Table 4: CPL logical interpretation of the different bivalent primitive model theoretic structures.

Given that feature bundles are interpreted conjunctively and we used a logic that only contains conjunction, this result is not all that surprising. That being said, the same logic was used to interpret the model theoretic structures defined with univalent primitives and there was only able to capture a privative feature system. This highlights the interaction between representation and logical interpretation. Depending on the representations used, different logics result in different outcomes.

Based on the results from this section we can come to a few conclusions. For example, the privative model defined over univalent primitives and interpreted with CPL logic is extensionally equivalent to the privative model defined over bivalent primitives and interpreted with CPL logic. That is, the set of natural classes that are defined from each model are identical. This suggests that it is the logical interpretation language that is doing most of the heavy lifting when modeling this type of feature system. The same thing can be said about the full feature systems, except it is a CNPL rather than CPL logical interpretation that is the important aspect of representing a full feature system. When it comes to contrastive feature systems, we see that it is in fact the representational aspect that is most important for ensuring that the model theoretic representation is in line with the feature table off of which it is based.

5 Discussion

The previous section showed how the combination of representational primitives and logical interpretation languages results in the ability to describe different types of feature systems. To be complete, we could also consider CNPL with bivalent primitives. Since using negation in the logic forces every feature to be binary, it should be no surprise that it is only the full system that can be correctly represented with this paring of primitives and logic. That being said, this would make it so negative features emerge from both the logic and the primitives which means there is a lot of redundancy built into the system.

So far, the discussion of 0 values has been focused on underspecification, but 0 is used for other things as mentioned earlier. One of the ways in which 0 values are used is for equipollent features such as the place features [labial], [coronal], [dorsal]. If these features are used in a full feature system, then it must be the case that they are interpreted as being binary. Consequentially, Coronal and ¬Coronal must exist as natural classes. It has sometimes been argued that [-coronal] is not a natural class (Yip, 1989). We can take away from this that in order to have any 0 values in a feature system, we cannot use negation in the interpretation language. This goes against most mathematical treatments of phonological features and natural classes (Keenan and Moss, 2009; Ojeda, 2013).

On the other hand, CNPL easily prevents any element from being both $+$ and $-$ for the same feature due to the law of excluded middle. It is logically impossible for any element x to satisfy both $F(x)$ and $\neg F(x)$. If we instead encode the $+$ and $-$ values directly into the primitives, there is nothing about the interpretation language that prevents any element x from satisfying both $+F(x)$ and $-F(x)$.

One option when evaluation $\mathcal{M}^\beta$ with CPL would be to specify feature cooccurrence restrictions. This would be a logical statement with subparts such as $\varphi(x) = \neg[+F(x) \vee -F(x)]$ which would be true only if a segment does not have both the positive and negative features. A model of a feature system $\mathcal{M}$ would therefore only satisfy φ if it did not allow for any segment to be both positive and negative for the same feature.

The goal of this paper was not to find the correct feature system. Rather, the goal was to see how to best represent each of the three different feature systems formally in order to better understand what the differences between each system are. Meaningful differences between the three systems do in fact emerge. For example, privative feature systems can be represented most simply as they minimally require univalent primitives and CPL logic. In order to describe a full feature system there needs to be either an increase in logical power (CNPL) or an increase in representational primitives (bivalent primitives). A contrastive feature system is the least flexible in how it can be represented as it requires CPL and bivalent primitives.

Deciding which of these are the "right" feature system cannot be directly decided based on the analysis provided in this paper. For example, a reviewer points out that most feature systems in use do use a combination of +,-, and 0 which would suggest that CPL with bivalent primitives is on the right path. This raises the question of what it means to be a minus feature in this type of system. If a minus feature is not a negated positive feature (its complement), then why use plusses and minuses at all? Answers to these types of questions lie beyond a purely formal account which is why the analysis given in this paper primarily provides a roadmap for future work on phonological feature systems and a better understanding of how to represent them in formal computational systems.

6 Conclusion

This paper used model theory and logic to explore three types of phonological feature systems commonly used in phonological analysis. The main takeaway is that using negation in the logical interpretation language (e.g., CNPL) forces every feature to be binary. Furthermore, in order to include non-binary oppositions in a feature system, the valuation of the features can be directly encoded into the primitives. One advantage of encoding feature valuation into the primitives is that it allows for the mixture of different types of oppositions without any noticeable issues. This opens the door for more inquiry into how phonological features can and should be viewed in a formal system.

7 Acknowledgments

I would like to thank Jeffrey Heinz, Charles Reiss, Karthik Durvasula, and members of the Stony Brook/Rutgers spring 2021 MathLing reading group for helpful comments and discussion on this material. I would also like to thank the anonymous reviewers for their constructive feedback.

References

Alan Bale and Charles Reiss. 2018. *Phonology: A formal introduction*. MIT Press.

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.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper & Row.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126(1):53–75.

San Duanmu. 2016. *A theory of phonological features*. Oxford University 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.

Edward Flemming. 2001. Scalar and categorical phenomena in a unified model of phonetics and phonology. *Phonology*, 18(1):7–44.

John Anton Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

Bruce Hayes. 2011. *Introductory phonology*. John Wiley & Sons.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Sharon Inkelas and Stephanie S Shih. 2016. Re-representing phonology: consequences of q theory. In *Proceedings of NELS*, volume 46, pages 161–174.

Roman Jakobson, C Gunnar Fant, and Morris Halle. 1951. *Preliminaries to speech analysis: The distinctive features and their correlates*. MIT press.

Adam Jardine, Nick Danis, and Luca Iacoponi. 2021. A formal investigation of q-theory in comparison to autosegmental representations. *Linguistic Inquiry*, 52(2):333–358.

C Douglas Johnson. 1972. *Formal Aspects of Phonological Description*. De Gruyter Mouton.

Edward L Keenan and Lawrence S Moss. 2009. *Mathematical structures in language*. CSLI.

Michael Kenstowicz and Charles Kisseberth. 1979. *Generative phonology: Description and theory*. Academic Press.

D Robert Ladd. 2014. *Simultaneous structure in phonology*, volume 28. OUP Oxford.

Leonid Libkin. 2004. *Elements of finite model theory*. Springer.

Connor Mayer and Robert Daland. 2020. A method for projecting features from observed sets of phonological classes. *Linguistic Inquiry*, 51(4):725–763.

John J McCarthy. 1988. Feature geometry and dependency: A review. *Phonetica*, 45(2-4):84–108.

Jeff Mielke. 2008. *The emergence of distinctive features*. Oxford University Press.

Chris Oakden. 2020. Notational equivalence in tonal geometry. *Phonology*, 37(2):257–296.

David Odden. 2005. *Introducing phonology*. Cambridge university press.

Almerindo E Ojeda. 2013. *A Computational Introduction to Linguistics: Describing Language in Plain PROLOG*. CSLI.

Charles Reiss. 2003. Quantification in structural descriptions: Attested and unattested patterns. *The Linguistic Review*, 20:305–338.

Donca Steriade. 1995. Underspecification and markedness. In John Goldsmith, editor, *The Handbook of Phonological Theory*, pages 114–174. Wiley-Blackwell.

Kristina Strother-Garcia. 2019. *Using Model Theory in Phonology: A Novel Characterization of Syllable Structure and Syllabification*. Ph.D. thesis, University of Delaware.

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*, volume 57 of *JMLR: Workshop and Conference Proceedings*, pages 66–78.

Nikolai Sergeevich Trubetzkoy. 1939. *Principles of phonology*. ERIC.

Mai Ha 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):1–26.

Moira Yip. 1989. Feature geometry and cooccurrence restriction. *Phonology*, 6(2):349–374.

A split-gesture, competitive, coupled oscillator model of syllable structure predicts the emergence of edge gemination and degemination

Francesco Burroni
Department of Linguistics
Cornell University
fb279@cornell.edu

Abstract

The phonological mechanisms responsible for the emergence of edge geminates in phonological processes like the Italian *Raddoppiamento (Fono-)Sintattico* (RS) are an open issue. Previous analyses of Italian treat gemination of (i) word initial consonants, (ii) morpheme-final consonants, and (iii) word final consonants as separate processes brought about by dedicated rule/constraints. We argue that these edge gemination processes result from the same, independently established principles. Through computational simulation of the split-gesture, competitive, coupled oscillator model of syllable structure of Articulatory Phonology, we show that increases in closure duration typical of geminates arise from changes to consonant/vowel couplings. Word initial gemination follows from coupling of a closure gesture to a preceding vowel across a word boundary. Word final gemination follows from coupling of a release gesture to a following vowel. In both cases, the posited structures reflect changes in syllabification hypothesized in previous work. The model simulation also predict different durations for resyllabified edge geminates and medial lexical geminates, in line with experimental findings on the topic. Changes to consonant/vowel couplings also account for the opposite effect: word initial degemination. Thus, the coupled oscillator model of Articulatory Phonology, originally developed to model intergestural timing, predicts the emergence of edge gemination/degemination.

1 Introduction

Word initial and word final geminates, collectively known as edge geminates, are employed contrastively in a highly restricted subset of the world's languages (Burroni and Maspong, To appear; Kraehenmann, 2011; Topintzi and Davis, 2017). This limited cross-linguistic distribution is often attributed to poor perceptual recoverability (Blevins, 2004). Despite the disfavorable phonetic characteristics of edge geminates, speakers of some languages productively create them in the speech stream as a result of regular phonological process. A well-known example is the so-called *Raddoppiamento* (*Fono-*)*Sintattico* (*RS*) in Central and Southern Italo-Romance varieties and Standard Italian (Passino, 2013 and references therein).

Edge-consonant gemination is not a unique feature of Italo-Romance. It has also been reported in a variety of typologically diverse and genetically unrelated languages (Bertinetto and Loporcaro, 1988), such as Finnish (Bertinetto, 1985), Biblical Hebrew (Lowenstamm, 1996), Pattani Malay (Paramal, 1991), Somali (Bertinetto and Loporcaro, 1988), Seri (Marlett and Stemberger, 1983), and Tamil (Ramasamy, 2011). Edge gemination is, thus, a phenomenon with clear cross-linguistic status, yet our understanding of it remains limited.

Three issues stand out in the discussion of edge geminates. The first issue is that, even though word initial gemination is by far the most widely studied case, other types of edge gemination also exist. Central and Southern Italian speakers, for instance, geminate initial consonants, as well as morpheme/word final consonants. Unified treatments of the phenomena have, however, rarely been pursued (for an exception *cf.* Passino, 2013; and partly Chierchia, 1986). Accordingly, the relationship among different types of edge gemination, if any, remains unclear.

The second issue is that phonological accounts represent derived initial geminates and medial lexical geminates with identical ambisyllabic

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 11-22.
Held on-line February 7-9, 2022

structures (Section 2). Crucially, there are systematic phonetic differences between the two. Edge geminates are consistently shorter than medial geminates, as experimental work on Italian shows (Payne, 2005; Campos-Astorkiza, 2012). These differences in duration are unexpected in current phonological accounts.

The third problem concerns the relationship between the emergence and loss of edge geminates. The emergence of edge geminates in Italian varieties and other languages has been analyzed as the synchronic consequence of regular phonological process. The loss of edge geminates, on the other hand, has been treated as a diachronic process as a consequence of perceptual/articulation biases and exemplar dynamics (Burroni and Maspong, To appear; Blevins and Wedel, 2009; Blevins, 2004). Nevertheless, synchronic degemination has been documented for Swiss German dialects (Kraehenmann and Jaeger, 2003) and synchronic diffusion of degemination has been documented for Pattani Malay (Burroni et al., 2020). Therefore, even though edge gemination and degemination share the basic property of altering consonantal duration, the mechanisms posited to account for them are remarkably different in both their motivation and timescale. No model of the relation between perceptual biases and changes in articulation has been developed either.

We argue that all types of edge gemination processes observed in languages like Central/Southern Italo-Romance varieties and Italian follow from changes to the dynamical coupling of consonants and vowels, which reflect changes in syllabification in a split-gesture, competitive, coupled oscillator model of syllable structure (Nam et al., 2009; Nam, 2007a; Nam, 2007c). This model also predicts the attested differences in duration between derived edge geminates and lexical medial geminates. Finally, changes in dynamical coupling between consonants and vowels also capture edge degemination, thus, providing a unified account of both phenomena.

2 Empirical phenomena under investigation and previous analyses

We investigate two set of empirical phenomena: (i) edge gemination in languages like Central/Southern Italo-Romance varieties and Italian and (ii) word initial degemination in languages like Swiss German and Pattani Malay. There are three different edge gemination processes in Italian.

First, speakers are known to produce new word initial geminates in the context of *RS,* provided that the target consonant is not already long. A word w$_i$ undergoes *RS* if: (i) the preceding word w$_{i-1}$ is stressed on the final syllable, e.g., /faˈrɔ ˈbɛne/ → [faˈrɔ ˈbːɛne] ‘I will do well’ and (ii) w$_{i-1}$ belongs to a closed class of monosyllables or disyllabic forms that do not have final stress but nonetheless trigger *RS*, e.g., /ˈkome ˈmaj/ → [ˈkome ˈmːaj] ‘how come’. Second, singleton word final codas, usually only present in loanwords, are geminated before a vowel initial suffix in morphological derivatives, e.g., /buldog/ + /-ino/ → [buldogːino] ‘small bulldog’. Third, word final codas are also geminated phrasally preceding another vowel initial word, e.g., /buldog agːresːivo/ → [buldogː agːresːivo] ‘aggressive bulldog’, a phenomenon often labeled *backwards RS*. Morpheme/word final gemination is subject to variation for final sonorants, especially [r], but it is categorical for obstruents (Passino, 2013). These three gemination phenomena are rarely offered a unified treatment, as the focus is usually on *RS* alone.

RS, the first type of gemination and the one that is most often treated in phonological work, is also subject to a fair amount of dialectal variation (Loporcaro, 1997), as, in some Italo-Romance varieties or regional pronunciations of Standard Italian, the process is triggered only after a small subset of lexical items or is absent altogether.

There are three main analyses of *RS*. The first approach holds that *RS* is a byproduct of well-formedness conditions on Italian (final) stressed rhymes or metrical feet. Under this approach, *RS* is due to speakers geminating a word initial consonant to create an ambisyllabic geminate. This ambisyllabic geminate makes a final stressed syllable closed and, thus, heavy, in conformity with a requirement that all Italian stressed syllables either have coda or contain a long vowel. A second approach holds that words that trigger *RS* contain an underlying, featurally empty consonantal slot that only surfaces *via* total assimilation in *RS* environments. Insertion of an entire CV skeleton has also been proposed to account for *RS*, and morpheme/word-internal gemination as well (Passino, 2013). A third approach holds that productive *RS* is limited to a post-tonic environment, accordingly, the only rule needed is a gemination rule of word initial consonants after word ending in stressed vowels.

None of these solutions is unproblematic. Well-formedness conditions on stress rhymes are at odds with the fact that *RS* also takes place after words that do not have final stress and that certain varieties also show no relationship between final stress and *RS* (Loporcaro, 1997). Empty consonant slots never surface and are, thus, problematic from an acquisition perspective. Additionally, there are words that trigger *RS* but did not have final consonants even when we look at the Latin ancestors of these words. Finally, reducing *RS* to a rule of onset gemination after final stressed vowels comes at the cost of greatly reducing the empirical coverage of the analysis, while certain assumptions regarding rule ordering are also necessary to prevent overapplication in contexts where stressed vowels do not trigger *RS*. All analyses agree that *RS* is produced by changes in syllabification, but they disagree on the rationale.

We show in the next sections that in a split-gesture coupled oscillator model of syllable structure all types of gemination follow purely from syllabification principles in a dynamical model, where no additional rationale is needed. We further show that this model predicts the observed phonetic differences between lexical medial geminates and derived edge geminates, a fact that is missed by other accounts.

The second set of phenomena we investigate are degemination processes. Degeminations of initial geminates has been reported after obstruent-final words in Swiss German, e.g., /s tːaŋk͡xə/ → [s taŋk͡xə] ‘the filling-up’ (Kraehenmann and Jaeger, 2003). Degeminations of initial geminates has also been reported for Pattani Malay, as some minimal pairs with and without initial geminates onsets are merging, e.g., [dapo] ‘kitchen’ and [dːapo] ‘at the kitchen’ are often no longer distinguishable in terms of closure duration of the initial consonant (Burroni et al., 2020). Degemination in Swiss German has been attributed to the loss of one of the two timing slots associated with initial geminates after obstruent-final words. The Pattani Malay neutralization has been analyzed in an exemplar model as a random walk in closure duration space leading to merger (Burroni and Maspong, To appear following Blevins and Wedel, 2009). In both cases a poor perceptual recoverability is invoked to drive change in the phonological representation of words, yet no link with the production of singletons and initial geminates has been explicitly proposed. We show that degemination also follows from changes in coupling reflecting changes in syllabification in a split-gesture, competitive, coupled oscillator model of syllable structure developed in the framework of Articulatory Phonology (Browman and Goldstein, 2000; Nam, 2007a).

3 The Articulatory Phonology split-gesture, competitive, coupled oscillator model of syllable structure

In the framework of Articulatory Phonology (AP) phonological primitives are identified with articulatory gestures. Gestures are conceptualized as time dependent driving forces that modify the value of tract variables (TVs) and the positions of the synergy of articulators associated with TVs. An example of Lip Aperture being driven until time 10 to a value of 0 mm, representing a bilabial closure [b], and until time 20 back to its original starting value of 10 mm, representing the release of the closure, is presented in Figure 1.

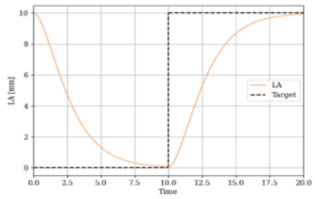


Figure 1 Example of Lip Aperture (LA) constriction and release, implemented with the model in Appendix A.

In the original Task Dynamic model of AP, the duration and relative timing of each gesture was considered part of the lexical representation of words and specified “by hand”. Browman and Goldstein (1990) later modelled the unfolding of a gesture in time with a “virtual” second order undamped systems that has the same stiffness of the original gestural system. The onset and target achievements of the gesture were arbitrarily identified with 0° and 240° of this virtual gestural cycle. Gestures could then be timed to each other by referring to phase relationships of their virtual cycles, e.g., synchronously (0° to 0°) or onset to target of the preceding gesture (0° to 240°). Other phase relationships were deemed possible, but the number of linguistically relevant ones was hypothesized to be highly constrained. Intergestural timing was modeled under a working assumption that, for n-gestures, at maximum $n-1$ local coupling forces between gestural pairs could be specified. All relative timing relationships could thus be defined in terms of coupling to a preceding gesture (Browman and Goldstein, 2000).

A more principled dynamical model of relative timing between two gestures was developed by Saltzman and Byrd (2000) using coupled oscillators. Saltzman and Byrd (2000) showed that punctate relative phases (or ranges of relative phases) can be generated by coupling the oscillators regulating the virtual gestural evolution cycles. The relative phase of two gestures is defined as the difference of their phases (ϕ) around the virtual unit cycle, i.e., $\psi = \phi_2 - \phi_1$. The relative phase task-space potential employed by Saltzman and Byrd (2000) is a simple cosine function:

$$V(\psi) = -a\, cos(\psi - \psi_0) \qquad (1)$$

In this model, a represent a parameter that controls how quickly target relative phase is achieved. ψ represents the current relative phase value of the system. ψ_0 represent a target relative phase. From this potential function a coupling force, defined as the negative of the derivative of the potential function is derived:

$$-\frac{dV(\psi)}{dt} = -\, a\, sin(\psi - \psi_0) \qquad (2)$$

The force function is added to each hybrid oscillator's equation to ensure that the coupled oscillators achieve the relative phase specified at the bottom of the potential valley and complete phase-locking, Appendix B. The coupled oscillator model developed by Saltzman and Byrd (2000) was extended to constellations larger than two gestures by Nam and Saltzman (2003), who challenged the assumption that gestures are timed locally to the preceding gesture. Following Browman and Goldstein (2000), Nam and Saltzman (2003) introduced the possibility of competitive coupling: several, mutually incompatible relative phase targets could now be specified for each pair of gestures. The consequence of competitive coupling is that surface relative timing among different gestures is a "compromise" of different relative phase equilibria specified for coupled oscillators. Nam and Saltzman (2003) focused on c-center timing, a non-local timing regime where the initiation of a word initial vowel gesture appears to be timed with the midpoint of the onset consonants forming a cluster. Nam and Saltzman (2003) showed that c-center, problematic for strictly local timing as it involves timing to an entire cluster, emerges spontaneously if competing relative phases are specified between two onset consonants and for each consonant to the vowel.

A full competitive model of syllable structure in Articulatory Phonology was developed by Nam (2007a; 2007b; 2007c). Nam proposed that the articulatory gestures associated with syllables can be represented as nodes in an undirected graph with no loops, where edges represent target phase couplings for the gestural nodes they connect. Using this graph representation, competing target relative phases can be specified for each gestural pair and competitive coupling is generalized to all possible gestural pairs.

A second feature of the model is that consonantal gestures were split into two gestures: a closure and a release gesture. Nam (2007b), following Browman (1994), argued that releases should be treated as separate gestures, rather than as a return to a neutral vocal tract position. The reason is that the stiffness and velocity of closures and releases are similar, thus, suggesting that both are actively controlled gestures. Nam (2007a; 2007c) also showed that vowels can display c-center timing to the midpoint of the closure and release of a single consonant onset, Figure 2. This another fact that can be taken as evidence for a multigestural representation of a single consonant, similar to that of clusters.

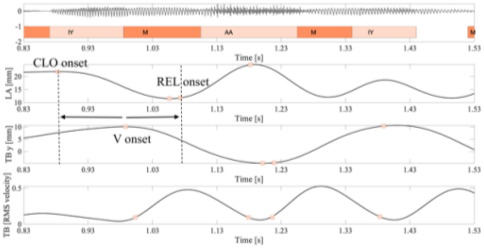


Figure 2 Electromagnetic articulography data exemplifying single consonant c-center for an English speaker producing the word *mommy*. Vowel onset is symmetrically displaced between closure and release.

C-center in singleton consonants has since been experimentally confirmed and further studied (Tilsen, 2017). Nam (2007a) also showed that many properties of phonological systems can be understood in a split-gesture model; among these are onset/coda asymmetries, both typological and developmental ones, and moraic structure and its acoustic reflexes. Notably, Nam (2007a) also hypothesized that many properties of geminates are best understood through the lenses of a split-gesture model.

4 The model

The model we present closely follows the one developed by Nam (2007a; 2007c). Syllables are represented as nodes in an undirected graph without loops. This graph is known as the coupling graph. Closure and release belonging to the same consonant are represented as separate nodes. An example of this split-gesture graph representation of CV and VC syllables is illustrated in Figure 3.

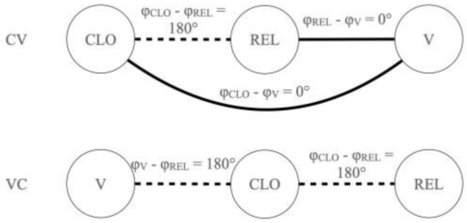

CV
CLO
$\varphi_{CLO} - \varphi_{REL} = 180°$
REL
$\varphi_{REL} - \varphi_V = 0°$
V
$\varphi_{CLO} - \varphi_V = 0°$
VC
V
$\varphi_V - \varphi_{REL} = 180°$
CLO
$\varphi_{CLO} - \varphi_{REL} = 180°$
REL

Figure 3 Split-gesture graph representation of CV (top) and VC (bottom) syllables, dashed lines represent anti-phase coupling, solid line in-phase

Figure 3 showcases another important constraint imposed on the model: only two target relative phases are assumed to be available: 0° and 180° (Tilsen, 2018). These are termed in-phase and anti-phase.

The rationale for only two phases is that only those are readily observed in the realm of human (and animal) movement, e.g., in transitions between different gaits of quadrupeds, like horses. Those two phase relationships have also been shown to emerge in experimental tasks involving rhythmic movement (Turvey, 1990). Other relative phase configurations can only be learned with training or emerge from competitive coupling (Nam, 2007a). In this model, the virtual cycle controlling the timing of each gesture is represented only in terms of phase around the unit circle. The differential equation controlling the evolution of each oscillator's phase in the system of coupled oscillators is defined as:

$$\dot{\theta_i} = \omega_i + \sum_{j=1}^{N} K\, sin(\theta_j - \theta_i - \psi_0) \quad (3)$$

ω_i is the natural frequency of the i^{th} oscillator, set to 2π to for our simulations. K is a coupling constant that determines the force exerted by each pair in settling towards target relative phase equilibria. θ_j with $j = 1 \dots n$ is the j^{th} oscillator's phase to which θ_i is coupled. ψ_0 is a relative phase target equilibrium for the relative phase of θ_i and θ_j. The model generalized Saltzman and Byrd's (2000) model to a larger system of oscillators. The matrix form of the model is presented in Appendix C.

This model returns $\dot{\theta}$, an $i \times 1$ vector of oscillator phases at each time step of the simulation of the differential equation. All differential equations were numerically integrated in MATLAB using a forward Euler method over a time range [0 100], the time step was fixed at .1. Following previous work (Nam, 2007a; Tilsen, 2018), the phase of each oscillator is mapped to a virtual gestural cycle using a cosine function. Gestures are hypothesized to be triggered once phase-locking is completed and the virtual cycle oscillator crosses 0°.

Following Tilsen (2018), we impose a constraint on initial phases such that each gestural oscillator has a higher initial phase than the gestural oscillator following it in the linearly ordered phonological sequence. For instance, for a CV sequence, with C split into CLO-REL, we impose a constraint $\varphi_{CLO} > \varphi_{REL} > \varphi_V$. These constraints on initial phase values are taken to be part of the lexical representation and to reflect learned order of movements (Tilsen 2018).

5 Experiments

5.1 Singleton c-center and geminate timing

The model can generate a variety of previously reported (relative) timing patterns.

Simulation of c-center timing is achieved by coupling the closure (CLO) and release (REL) oscillators with a target relative phase of 180° (anti-phase), while both CLO/vowel (V) and REL/V are coupled with a target relative phase of 0° (in-phase). The results are the stable relative phase patterns displayed in Figure 4 top and middle. CLO and REL have a relative phase of 120°, while CLO and V and REL and V have a relative phase of 60° and -60° respectively.

The model, thus, predicts a symmetric initiation of the V gesture after the initiation of CLO and before the initiation of REL, Figure 4 bottom. Arrows depict the initiation of each gesture after oscillators have settled in stable relative phases.

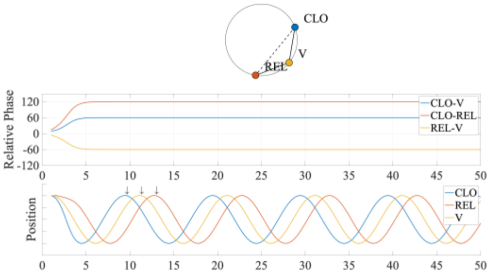


Figure 4 Simulation of single consonant c-center.

As is well-known, lexical geminates differ from singletons are in terms of a longer closure duration (Ladefoged and Maddieson, 1996). In a split-gesture model geminates are represented by an increased relative timing between the initiation of the CLO and REL gestures of the same consonant. Since the CLO and REL of a consonant control the same TV, a later initiation of the REL means that CLO will have control of the articulators for a longer period of time. Nam (2007a) suggested that this can be achieved by assuming that the CLO and REL oscillators are anti-phase coupled, like in a singleton, but, crucially, only the REL oscillator is in-phase coupled to the V oscillator. The result of this coupling is complete anti-phase between CLO and REL/V oscillators. This relative phase pattern predicts a longer delay in the initiation of the REL compared to the initiation of CLO, consistent with the longer durations of geminates, Figure 5.

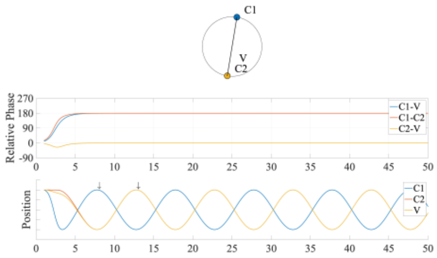


Figure 5 Simulation of geminate timing.

Recent experimental work has shown that the relative timing of closure and vowel initiation for medial geminates is stable across different speech rates (Tilsen and Hermes, 2020). This suggests a stable timing relationship, i.e., in-phase, between the two. Accordingly, a better representation for geminates in a split-gesture model may be coupling only CLO to V, while maintaining anti-phase coupling for CLO and REL. Under this coupling, the result is CLO and V stabilizing in-phase to each other and in anti-phase with REL, Figure 6.

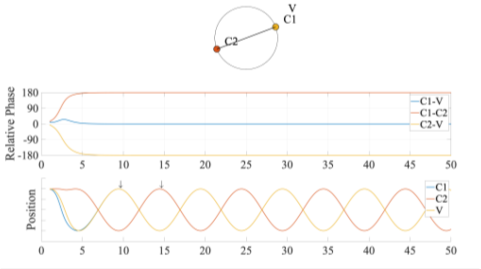


Figure 6 Simulation of geminate timing revised

5.2 Word initial Gemination

Following previous work (Section 2), we subscribe to the idea that *RS* is a change in syllabification. In particular, *RS* is the formation of an ambisyllabic geminate that acts as both a coda of the preceding syllable and as a word initial onset, as envisioned in all previous analyses. No further dedicated mechanism is necessary for the emergence of word initial geminates. The creation of an ambisyllabic geminate is conceptualized in dynamical terms as follows. The emergence of a new coda amounts to coupling the oscillators of a word final V and a word initial CLO gesture in anti-phase and to decoupling the CLO oscillator from the following V oscillator. No change ensues between the coupling of the CLO and REL oscillators of the word initial consonant, as they still have a target anti-phase relationship. We also assume that the final vowel of the word triggering *RS* and the first vowel of the word undergoing *RS* are anti-phase coupled or sequential. The coupling graph is illustrated in Figure 7.

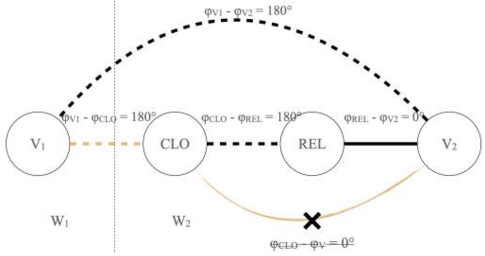


Figure 7 Proposed coupling graph for *RS*.

If we implement these target relative phases in the model, the result is the achievement of a target relative phases between CLO and REL of 135°. This relative phase relationship ensures that the CLO has active control of the TV for a period that is longer than for singleton (120°), but shorter than for lexical geminates (180°), in line with findings showing that *RS* derived edge geminates closure duration is not as long as that of lexical medial geminates. We return to this issue in Section 6. The model further predicts the correct relative timing initiation: final V of the word triggering *RS*, followed by CLO, followed by REL, followed by V2 of word undergoing *RS*, Figure 8.

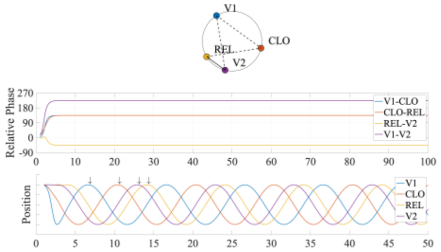


Figure 8 Simulation of *RS*.

5.3 Word final gemination across morpheme/word boundaries

Following previous work detailed in Section 2, especially Passino (2013), we assume that word final gemination across morpheme/word boundaries follows from changes in syllabification, just like *RS*. In this case, the morpheme or word final consonant of the word or stem triggering gemination becomes ambisyllabic and hence geminates, like in *RS*. In dynamical terms, a coupling relationship between the oscillators of a word final REL and a word initial V2 gesture emerges, while the REL oscillator is no longer coupled to the preceding V1 oscillator, as is usually the case for codas that share a mora with preceding vowels and shorten them (Nam, 2007c). No change ensues for the coupling of the CLO and REL oscillators of the word final consonant. They still have a target anti-phase relationship. The coupling graph is illustrated in Figure 9.

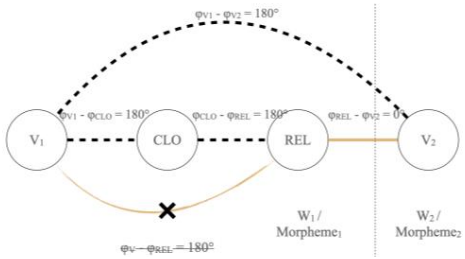


Figure 9 Proposed coupling graph for word/morpheme final gemination.

Exactly as for *RS*, the model predicts a target relative phase between CLO and REL of 135°, Figure 10. The model, thus, generates both the correct relative timing pattern and it also predicts word final gemination across morpheme/word boundary. Again, derived edge geminates are expected to be shorter than lexical medial geminates.

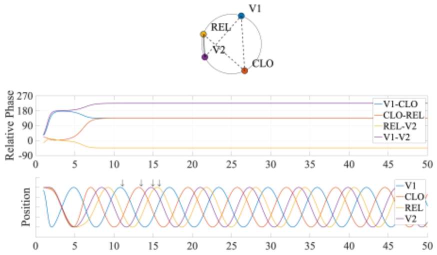


Figure 10 Simulation of word/morpheme final gemination.

5.4 Experiment 4: word initial degemination

In languages like Swiss German and Pattani Malay, synchronic or lexically diffusing degemination of word initial geminates has been observed in experimental work, Section 2. These have been attributed to poor perceptual recoverability triggering changes in phonological representation or in exemplar dynamics of closure duration. Yet, the relationship between degemination and articulation has been left unaddressed. In a split-gesture, competitive, coupled oscillator model of syllable structure incipient degemination can be captured simply as the emergence of a more stable structure where both CLO and REL are in phase coupled to V. Lexical initial geminates are represented with a coupling graph identical to lexical medial geminates: in-phase CLO-V and antiphase CLO-REL. The change in coupling graph structure that triggers degemination is the emergence of a stable in-phase coupling between REL and V, Figure 11.

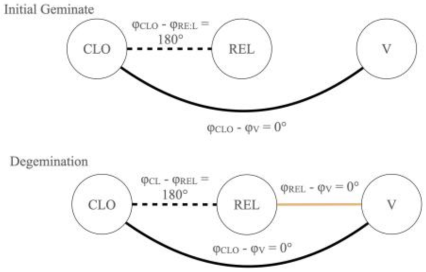


Figure 11 Coupling graphs for initial geminates and word initial degemination.

Obviously, if this coupling graph is used as input to the model, c-center timing emerges. The result is a relative phase between CLO and REL of 120°, identical to that of singletons. Thus, the result of this change in coupling structure is degemination, as suggested by Nam (2007a), Figure 12.

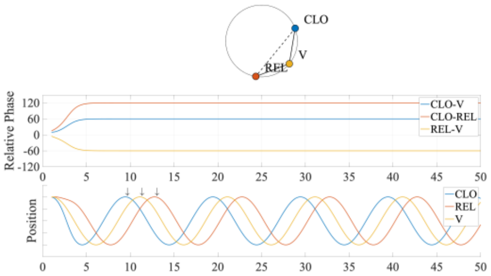


Figure 12 Simulation of word-initial degemination.

5.5 Dynamics of syllabification as the main force behind edge (de)gemination

Having illustrated how the model predicts the emergence and loss of edge geminates, we can now fully appreciate the rationale behind these phenomena: dynamical principles of syllabification in a competitive, coupled-oscillator model of intergestural timing.

The gemination phenomena we have discussed follow only from translating previously hypothesized changes in syllabification into changes to coupling graphs of articulatory gestures forming consecutive syllables. *RS* and morpheme/word final gemination had already been hypothesized to result from the phonological requirement of creating an ambisyllabic geminate; either to create heavy syllables or because of assimilation, originally due to empty consonants (Section 2). In the spirit of dynamical system theory, the model we have presented does not force to choose between these competing alternatives. Instead, the gemination has no further rationale: the process follows purely from the emergence of new dynamical couplings among articulatory gesture. These changes reflect resyllabification near a word boundary, where coupling strengths have long been hypothesized to be weak and gestural sliding has been observed (Browman and Goldstein, 2000).

We can, thus, hypothesize that resyllabification emerges in speech production as a consequence of various factors, e.g., fluctuations in coupling strength due to noise in the production system or because of the effects of speech rate. Once resyllabification alters the dynamical couplings, edge gemination is the natural response of the phonological system. No dedicated rule of edge gemination is needed.

The case of edge degemination follows from slightly different principles. It is not a case of resyllabification across a morpheme/word boundary, but, rather, it represents the emergence of a less marked coupling graph. In other words, it represents a more stable syllabic configuration. Specifically, the emergence of a new coupling between CLO and V, that triggers edge degemination (Figure 11 Coupling graphs for initial geminates and word initial degemination.), represents the emergence of a coupling graph where both articulatory gestures forming a consonant are timed to the vowel. Such configurations with a higher number of links, together with the emergence of in-phase relationships, have been demonstrated to lead to syllable productions that are less sensitive to the effects of noise (Nam, 2007a).

In sum, the model we have presented shows that the emergence and loss of edge geminates are tightly linked as the byproduct of changes to coupling graphs that reflect resyllabification and more stable syllabic configurations.

6 Discussion

We have demonstrated that changes in dynamical couplings, reflecting syllabification, can be responsible for the emergence of (i) word initial gemination, (ii) word/morpheme final gemination, and (iii) word initial degemination. The changes in syllabification were implemented by introducing changes in the dynamical coupling between the oscillator controlling the relative timing of CLO, REL, and V in a split-gesture, competitive, coupled oscillator model of syllable structure. This model offers a unified theory of the articulatory features that accompany the emergence and disappearance of edge geminates.

Furthermore, the model also predicts durational difference between derived edge geminates and lexical medial geminates. This is accomplished by different phase locking patterns: for lexical (medial) geminates the CLO and REL oscillators stabilize at a relative phase of 180°; for derived edge geminates the relative phase is 135°. Recall that the difference between singleton, displaying c-center timing, and geminates is one of 120° vs 180°. Accordingly, edge geminates only cover ¼ (15°/60°) of the relative phase difference that separates singleton from geminates. This relative phase patterns are compatible with the experimental findings of Campos-Astorkiza

(2012), who reported that geminates derived via *RS* have a percentage of lengthening, compared to singleton, in the range of 23-60% (on average around 50%). For lexical geminates the range is 200-276%. The model presented, thus, offers not only unified treatment of different types of edge gemination and degemination, but it also predicts phonetic differences between derived initial and lexical medial geminates that align with experimental findings. Crucially, the model does not require any dedicated mechanism to accomplish this, the phonological processes follow purely from dynamical couplings that reflect changes in syllabification. In this way, shared intuitions presented in previous work can be unified without a need for choosing any one rationale, as the system is self-organizing.

The model also has some limitations. First, it accounts for difference between singleton and geminates purely in terms of relative intergestural timing. However, differences between singletons and geminate are likely to be manifested also in intragestural timing due to differences in parameters like targets, stiffnesses, *etc*. Furthermore, translating relative timing into periods of gestural activation intervals is a non-trivial problem, for which a variety of solutions have been proposed (Tilsen, 2018).

A second limitation is that recent experimental evidence (Tilsen and Hermes, 2020) has shown that the onset of geminate release, with respect to either the onset of the closure or the vowel, is linearly delayed as speech rate increases. For singletons the relative timing patterns are relatively unaffected. Tilsen and Hermes (2020) interpreted these different timing regimes as evidence that singletons can be modelled with coupled oscillators, but competitive queuing and feedback based gestural suppressions (Tilsen, 2016) may be necessary to generate the geminate timing patterns.

This is a more general problem of the coupled oscillator model and of the TD model that regulates gestural evolution. They are feedforward systems with no feedback. This assumption is clearly problematic for speech (Shaw and Chen, 2019; Tilsen, 2016; Parrell et al., 2019). Accordingly, scholars have proposed extensions of the model that take feedback into account (Tilsen, 2016; Parrell et al., 2019). Integrating feedback mechanisms for different types of geminates is a direction that needs to be further explored.

The coupled oscillator model is also sensitive to the initial conditions of the simulation. Specifically, it is sensitive to the initial phase of each gestural oscillator. To side step this problem, we have imposed constraints on initial phases that we take to a be a reflex of lexical representation and linear ordering (Tilsen, 2018). However, these constraints may betray the need for integrating a competitive queuing model on top of a coupled oscillator model of syllable structure (Tilsen 2016; 2018).

Finally, the coupling structures posited to account for the emergence and disappearance of edge geminates need empirical verification *via* collection of articulatory data, e.g., EMA or real time MRI. Such a dataset may also be a starting point to explore how the creation of new dynamical couplings may emerge in the first place. In particular, we can hypothesize that fluctuations in coupling strength may give rise to trial to trail variability in coupling of consonants at word edges and vowels (Browman and Goldstein, 2000). Ultimately, these changes may be phonologized as changes to coupling graphs. This hypothesis, however, requires empirical testing.

7 Conclusion

We have demonstrated that the AP split-gesture, competitive, coupled oscillator model provides us with a self-organizing model of syllable structure where edge-gemination and degemination emerge from dynamical coupling of closure and release oscillators with vowel oscillators. The model offers a unified analysis of different types of edge gemination and degemination, an aspect that was missing in previous phonological work. Moreover, the model also predicts crucial phonetic differences between derived edge geminates and lexical medial geminates reported in experimental work, but missing in previous phonological analyses. In sum, the coupled oscillator model of Articulatory Phonology, originally designed to model intergestural timing, has proven to be successful at predicting the finer details of elusive phonological processes like edge gemination and degemination.

Acknowledgements

I am indebted to the members of the Cornell Phonetics Lab, Sam Tilsen, and three anonymous reviewers for providing useful feedback on earlier versions of this paper.

References

Pier Marco Bertinetto. 1985. A proposito di alcuni recenti contributi alla prosodia dell'italiano. *Annali della Scuola normale superiore di Pisa. Classe di lettere e filosofia,* 15(2):581–643.

Pier Marco Bertinetto and Michele Loporcaro. 1988. On empty segments and how they got that way. In *Certamen phonologicum, Papers from the 1987 Cortona Phonology Meeting*, pages 37–62. Rosenberg and Sellier, Turin.

Juliette Blevins. 2004. *Evolutionary phonology: The emergence of sound patterns*. Cambridge University Press.

Juliette Blevins and Andrew Wedel. 2009. Inhibited sound change: An evolutionary approach to lexical competition. *Diachronica*, 26(2):143–183.

Catherine P Browman. 1994. Lip aperture and consonant releases. In *Phonological Structure and Phonetic Form. Papers in Laboratory Phonology III*, pages 331–353. Cambridge University Press,.

Catherine P Browman and Louis Goldstein. 2000. Competing constraints on intergestural coordination and self-organization of phonological structures. *Les Cahiers de l'ICP. Bulletin de la communication parlée*(5):25–34.

Francesco Burroni and Sireemas Maspong. To appear. Re-examining Initial Geminates: Typology, Evolutionary Phonology, and Phonetics. In *Historical Linguistics 2019*. John Benjamins.

Francesco Burroni, Sireemas Maspong, Pittayawat Pittayaporn, and Pimthip Kochaiyaphum. 2020. A new look at Pattani Malay Initial Geminates: a statistical and machine learning approach. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 21–29.

Rebeka Campos-Astorkiza. 2012. Lengthening and prosody in Tuscan Italian. *Anuario del Seminario de Filología Vasca" Julio de Urquijo": International journal of basque linguistics and philology*, 46(1):83–108.

Gennaro Chierchia. 1986. Length, syllabification, and the phonological cycle in Italian. *Italian Journal of Linguistics*, 8(1):5–34.

Astrid Kraehenmann. 2011. Initial geminates. *The Blackwell companion to phonology*:1–23.

Astrid Kraehenmann and Marion Jaeger. 2003. Phrase-initial geminate stops: articulatory evidence for phonological representation. In *Proceedings of the 15th International Conference of the Phonetic Sciences*, pages 2725–2728, Barcelona.

Peter Ladefoged and Ian Maddieson. 1996. *The sounds of the world's languages*.volume 1012. Blackwell Oxford.

Michele Loporcaro. 1997. *L'origine del raddoppiamento fonosintattico: saggio di fonologia diacronica romanza*.volume 115. Francke A. Verlag.

Jean Lowenstamm. 1996. The beginning of the word. In *Phonologica 1996 Syllables!?: Proceedings of the 8th International Phonology Meeting*, pages 153–166, The Hague. Holland Academic Graphics.

Stephen A Marlett and Joseph Paul Stemberger. 1983. Empty consonants in Seri. *Linguistic Inquiry*, 14(4):617–639.

Doris Mücke, Anne Hermes, and Sam Tilsen. 2020. Incongruencies between phonological theory and phonetic measurement. *Phonology*, 37(1):133–170.

Hosung Nam. 2007a. *A gestural coupling model of syllable structure*. Ph.D. thesis, Yale.

Hosung Nam. 2007b. Articulatory modeling of consonant release gesture. In *International Congress on Phonetic Sciences XVI*, pages 625–628.

Hosung Nam. 2007c. Syllable-level intergestural timing model: Split-gesture dynamics focusing on positional asymmetry and moraic structure. In *Laboratory Phonology 9*, pages 483–506, Urbana Champaign. Mouton De Gruyter. publisher: Mouton de Gruyter Berlin.

Hosung Nam, Louis Goldstein, and Elliot Saltzman. 2009. Self-organization of syllable structure: A coupled oscillator model. *Approaches to phonological complexity*, 16:299–328.

Hosung Nam and Elliot Saltzman. 2003. A competitive, coupled oscillator model of syllable structure. In *Proceedings of the 15th international congress of phonetic sciences*, volume 1.

Waemaji Paramal. 1991. *Long consonants in Pattani Malay: The result of word and phrase shortening*. PhD Thesis, Faculty of Graduate Studies, Mahidol University.

Benjamin Parrell, Vikram Ramanarayanan, Srikantan Nagarajan, and John Houde. 2019. The FACTS model of speech motor control: Fusing state estimation and task-based control. *PLoS computational biology*, 15(9):e1007321.

Diana Passino. 2013. A unified account of consonant gemination in external sandhi in Italian: Raddoppiamento Sintattico and related phenomena. *The Linguistic Review*, 30(2):313–346.

Elinor M Payne. 2005. Phonetic variation in Italian consonant gemination. *Journal of the International Phonetic Association*, 35(2):153–181.

Mohana Dass Ramasamy. 2011. *Topics in the morphophonology of standard spoken Tamil (sst): An Optimality Theoretic study*. PhD Thesis, Newcastle University.

Elliot Saltzman and Dani Byrd. 2000. Task-dynamics of gestural timing: Phase windows and multifrequency rhythms. *Human Movement Science*, 19(4):499–526.

Elliot Saltzman and Kevin Munhall. 1989. A dynamical approach to gestural patterning in speech production. *Ecological psychology*, 1(4):333–382.

Jason A Shaw and Wei-rong Chen. 2019. Spatially conditioned speech timing: Evidence and implications. *Frontiers in psychology*, 10:2726.

Sam Tilsen. 2016. Selection and coordination: The articulatory basis for the emergence of phonological structure. *Journal of Phonetics*, 55:53–77.

Sam Tilsen. 2017. Exertive modulation of speech and articulatory phasing. *Journal of Phonetics*, 64:34–50.

Sam Tilsen. 2018. Three mechanisms for modeling articulation: selection, coordination, and intention. *Ithaca, NY: Cornell University*.

Sam Tilsen and Anne Hermes. 2020. Nonlinear effects of speech rate on articulatory timing in singletons and geminates. In *12th International Seminar on Speech Production*.

Nina Topintzi and Stuart Davis. 2017. On the weight of edge geminates. *The phonetics and phonology of geminate consonants*, 2:260–282.

Michael T Turvey. 1990. Coordination. *American psychologist*, 45(8):938.

Appendix A: The Task Dynamic Model

In the Task Dynamic model the state of each TV is represented as a second order critically damped oscillatory system, following the Task Dynamic (TD) approach to motor control in speech (Saltzman and Munhall, 1989)

$$m\ddot{x} + b\dot{x} + k(x - T(t)) = 0$$

m represents the articulator mass. It is usually ignored and set to the unit value. b represents the damping coefficient. Critical damping, $b = 2\sqrt{km}$, is assumed to enforce asymptotic target achievement without oscillations. k represents the stiffness parameter, which determines how quickly the target state of the system is achieved. Higher stiffness corresponds to a quicker target achievement. Finally, x and *T(t)* represent the current positional value of the system and its target state, respectively.

Appendix B: The hybrid oscillator model of Saltzman and Byrd (2000)

In the original coupled oscillator model of oscillators Saltzman and Byrd (2000) the force function is transformed into a task specific coupling force that drives changes in the acceleration of a hybrid oscillator that arises from the combination of a Van Der Pol and Rayleigh limit cycle

$$\ddot{x} = -\alpha\dot{x} - \beta x^2\dot{x} - \gamma\dot{x}^3 - {\omega_0}^2 x$$

α represents a linear damping term, while β and γ non-linear van der Pol and Rayleigh damping, respectively. ω_0 represents the oscillator natural frequency.

Appendix C: Matrix implementation of the split-gesture, competitive, coupled oscillator model of syllable structure of Articulatory Phonology

The differential equation controlling the system of oscillators in our model is:

$$\dot{\theta} = \omega + \sum_{j=1}^{N} K\, sin\left(A_j \circ \Phi_j^T - A_j \circ \Phi_j - \Psi_{0_j}\right)$$

ω is the natural frequency of each oscillator and it is hypothesized to be identical for each oscillator, following previous work (Nam, 2007a) . Φ is an is $i \times j$ ($i = n, j = n$, where n is the number of oscillators in the system) matrix of initial phases for each oscillator i, with the value repeated across columns. A is an $i \times j$ adjacency matrix such that its element a_{ij} is defined as 1 if the oscillator i is coupled with oscillator j, and 0 otherwise. Ψ_{0_j} is an $i \times j$ matrix of target relative phase where each cell $\psi_{0_{ij}}$ represents a target relative phase for the oscillator pair θ_i and θ_j. If the oscillators are uncoupled the target relative phase is set to 0. K is a matrix of coupling constants. It is set to a unit matrix in all simulations reported to avoid exploding the parameter space, it could however be used to model cross-linguistic differences (Mücke et al., 2020).

ANLIzing the Adversarial Natural Language Inference Dataset

Adina Williams, Tristan Thrush, Douwe Kiela
Facebook AI Research
{adinawilliams, tthrush, dkiela}@fb.com

Abstract

We perform an in-depth error analysis of the Adversarial NLI (ANLI) dataset, a recently introduced large-scale human-and-model-in-the-loop natural language inference dataset collected dynamically over multiple rounds. We propose a fine-grained annotation scheme for the different aspects of inference responsible for the gold classification labels, and use it to hand-code the ANLI development sets in their entirety. We use these annotations to answer a variety of important questions: which models have the highest performance on each inference type, which inference types are most common, and which types are the most challenging for state-of-the-art models? We hope our annotations will enable more fine-grained evaluation of NLI models, and provide a deeper understanding of where models fail (and succeed). Both insights can guide us in training stronger models going forward.

1 Introduction

Natural Language Inference (NLI) is one of the canonical benchmark tasks for research on Natural Language Understanding (NLU). NLI[1] has characteristics that make it desirable both from theoretical and practical standpoints. Theoretically, entailment is, in the words of Richard Montague, “the basic aim of semantics” (Montague, 1970, p. 223 fn.), and indeed meaning in formal semantics relies on necessary and sufficient truth conditions. Practically, NLI is easy to evaluate and intuitive even to non-linguists, enabling data to be collected at scale with crowdworker annotators. Moreover, many core NLP tasks can also easily be converted to NLI problems (White et al. 2017; Demszky et al. 2018; Poliak et al. 2018a i.a.) suggesting that NLI is generally seen as a good proxy for measuring models’ overall NLU capabilities.

Benchmark datasets are essential for driving progress in NLP and machine learning (DataPerf Working Group, 2021). In recent years, large-scale NLI benchmarks like SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) have established a straightforward basis for comparison between trained models. However, with the advent of transformer language models, many benchmarks are now reaching saturation, leading some to wonder: have we solved NLI and, perhaps, NLU? However, the recent ANLI dataset (Nie et al., 2020a) illustrated that our models do not yet perform NLI in the robust and generalizable way that humans can. In this paper we ask: where do our models still fall short?

To improve towards general NLU, merely listing examples of failure cases is not by itself sufficient. We also need a quantifiable and finer-grained understanding of *which phenomena are responsible* for failures (or successes). Since the dynamic adversarial set up of ANLI encouraged human annotators to exercise their creative faculties to fool model adversaries, the data contains a wide range of possible inferences (as we will show). Because of this, ANLI is an ideal testbed for studying current model shortcomings, and for characterizing what future models will have to do in order to make progress on the NLI task.

Towards that end, we propose a genre-agnostic annotation scheme for NLI that classifies example pairs into 40 inference types. It is hierarchical, reaching a maximum of four levels deep, enabling analysis of model performance at a flexible level of granularity. We also contribute expert hand-annotations on the ANLI development sets (3200 sentence pairs) according to our scheme[2], thereby extending the usefulness of the ANLI dataset by making it possible to analyze future models. We

[1] Also known as recognizing textual entailment (RTE; Fyodorov et al. 2000; Dagan et al. 2006, i.a.).

[2] All annotations are publicly available at https://github.com/facebookresearch/anli/anlizinganli.

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 23-54.
Held on-line February 7-9, 2022

Context	Hypothesis	Rationale	Gold/Pred. (Valid.)	Tags
Eduard Schulte (4 January 1891 in Düsseldorf – 6 January 1966 in Zürich) was a prominent German industrialist. He was one of the first to warn the Allies and tell the world of the Holocaust and systematic exterminations of Jews in Nazi Germany occupied Europe.	Eduard Schulte is the only person to warn the Allies of the atrocities of the Nazis.	The context states that he is not the only person to warn the Allies about the atrocities committed by the Nazis.	C/N (CC)	Tricky, Prag., Numerical, Ordinal
Kota Ramakrishna Karanth (born May 1, 1894) was an Indian lawyer and politician who served as the Minister of Land Revenue for the Madras Presidency from March 1, 1946 to March 23, 1947. He was the elder brother of noted Kannada novelist K. Shivarama Karanth.	Kota Ramakrishna Karanth has a brother who was a novelist and a politician	Although Kota Ramakrishna Karanth's brother is a novelist, we do not know if the brother is also a politician	N/E (NEN)	Basic, Coord., Reasoning, Plaus., Likely, Tricky, Syntactic
Toolbox Murders is a 2004 horror film directed by Tobe Hooper, and written by Jace Anderson and Adam Gierasch. It is a remake of the 1978 film of the same name and was produced by the same people behind the original. The film centralizes on the occupants of an apartment who are stalked and murdered by a masked killer.	Toolbox Murders is both 41 years old and 15 years old.	Both films are named Toolbox Murders one was made in 1978, one in 2004. Since it is 2019 that would make the first 41 years old and the remake 15 years old.	E/C (EE)	Reasoning, Facts, Numerical Cardinal, Age, Basic, Coord., Tricky, Wordplay

Table 1: Examples from development set. 'corr.' is the original annotator's gold label, 'pred.' is the model prediction, 'valid.' is the validator label(s).

find that examples requiring models to resolve references, utilize external knowledge, and deploy syntactic abilities remain especially challenging. Our annotations are publicly available, and we hope they will be useful for benchmarking progress on particular inference types and exposing weaknesses of future NLI models.

2 Background

We proposes an inference type annotation scheme for the Adversarial NLI (ANLI) dataset, which was collected via a gamified, adversarial human-and-model-in-the-loop format using the Dynabench platform (Kiela et al., 2021; Ma et al., 2021). Human annotators are matched with a **target model** trained on existing NLI data, and tasked with finding examples that fooled it into predicting the wrong label. Dynamically collecting data has since been shown to have training-time benefits above statically collected data (Wallace et al., 2021). Other than being dynamic, ANLI was collected with a similar method to SNLI and MNLI: untrained crowdworkers are given a context—and one of three classification labels, i.e., Entailment, Neutral and Contradiction—and asked to write a hypothesis. Table 1 provides examples.

The ANLI dataset was collected in English over three rounds, with different target model adversaries each round. The first round adversary was a BERT-Large (Devlin et al., 2019) model trained on SNLI and MNLI. The second was a RoBERTa-Large (Liu et al., 2019) ensemble trained on SNLI and MNLI, as well as FEVER (Thorne et al., 2018) and the training data from the first round. The third round adversary was a RoBERTa-Large ensemble trained on all previous data, plus the training data from the second round, with the additional difference that the contexts were sourced from multiple domains (rather than just from Wikipedia, as in the preceding rounds). The ANLI dataset is split so that all development and test set data were human-validated as model-fooling.

The ANLI dataset creators encouraged crowdworkers to give free rein to their creativity (Nie et al., 2020a, p.8).[3] Annotators explored, then ultimately converged upon, inference types that challenged each round's target model adversary. For example, the target model in round 1 was often fooled by numbers (see §4), which means the development set from round 1 (i.e., A1) contains many NUMERICAL examples. Training a later rounds' adversary on A1 then should result in a model that does better on such examples. Ultimately, crowdworkers would be less successful at fooling later adversaries with numbers, and fewer NUMERICAL examples will end up in later development sets.[4] In this way, understanding how inference types dynamically shift across the ANLI development sets can illuminate the capabilities of the target models used to collect them.

3 Developing A Scheme for Annotating Types of Inferences in NLI

Categorizing sentential inference relations into types is by no means a new endeavor (see the Doctrine of Categories from Aristotle's *Organon*): ample research has aimed to understand model behavior and/or develop best annotation practices which ought to be incorporated. However, a scheme should be, at least to some extent, tai-

[3] Gamification generally results in wide coverage datasets (Joubert et al., 2018; Bernardy and Chatzikyriakidis, 2019).

[4] Assuming that models trained on later rounds don't suffer from catastrophic forgetting.

Top Level	Second Level	Description
Numeral	Cardinal	basic cardinal numerals (e.g., 56, 57, 0, 952, etc.).
	Ordinal	basic ordinal numerals (e.g., 1^{st}, 4^{th}, 72^{nd} etc.).
	Counting	counting references in the text, such as: *Besides A and B, C is one of the monasteries located at Mt. Olympus.* ⇒ *C is one of three monasteries on Mount Olympus.*
	Nominal	numbers as names, such as: *Player 37 scored the goal* ⇒ *a player was assigned jersey number 37*.
Basic	Comp.& Super.	degree expressions denoting relationships between things, such as: *if X is faster than Y* ⇒ *Y is slower than X*
	Implications	cause and effect, or logical conclusions that can be drawn from clear premises. Includes classical logic types such as Modus Ponens.
	Idioms	idioms or opaque multiword expressions, such as: *Team A was losing but managed to beat the other team* ⇒ *Team A rose to the occasion*
	Negation	inferences relying on negating content from the context, with “no”, “not’, “never”, “un-” or other linguistic methods
	Coordinations	inferences relying on “and”, “or”, “but”, or other coordinating conjunctions.
Ref.	Coref.	accurately establishing multiple references to the same entity, often across sentences, such as: *Sammy Gutierrez* is *Guty*
	Names	content about names in particular (e.g., *Ralph is a male name*, *Fido is a dog’s name*, *companies go by acronyms*)
	Family	content that is about families or kinship relations (e.g., *if X is Y’s aunt, then Y is X’s nephew/niece and Y is X’s parent’s sibling*)
Tricky	Syntactic	argument structure alternations or changes in argument order (e.g., *Bill bit John* ⇒ *John got bitten.*, *Bill bit John* ⇏ *John bit Bill*)
	Pragmatic	presuppositions, implicatures, and other kinds of reasoning about others’ mental states: *It says ‘mostly positive’ so it stands to reason some were negative.*
	Exhaustification	pragmatic reasoning where all options not made explicit are impossible, for example: *a field involves X, Y, and Z* ⇒ *X, Y and Z are the only aspects of the field*
	Translation	examples with text in a foreign language or using a foreign orthography.
	Wordplay	puns, anangrams, and other fun language tricks, such as *Margaret Astrid Lindholm Ogden’s initials are MALO, which could be scrambled around to form the word ’loam’.*
Reasoning	Plausibility	the annotators subjective impression of how plausible a described event is (e.g. *Brofiscin Quarry is named so because a group of bros got together and had a kegger at it.* and *Fetuses can’t make software* are unlikely)
	Facts	common facts the average human would know (like that the year is 2020), but that the model might not (e.g., *the land of koalas and kangaroos* ⇒ *Australia*), including statements that are clearly not facts (e.g., *In Ireland, there’s only one job.*)
	Containment	references to mereological part-whole relationships, temporal containment between entities (e.g., *October is in Fall*), or physical containment between locations or entities (e.g., *Germany is in Europe*). Includes examples of bridging (e.g., *the car had a flat* ⇒ *The car’s tire was broken*).
Imperfections	Error	examples for which the expert annotator disagreed with the gold label, such as the gold label of neutral for the pair *How to limbo. Grab a long pole. Traditionally, people played limbo with a broom, but any long rod will work* ⇒ *limbo is a type of dance*
	Ambig.	example pairs for which multiple labels seem to the expert to be appropriate. For example, with the context *Henry V is a 2012 British television film*, whether *Henry V is 7 years old this year* should get a contradiction or neutral label depends on what year it is currently as well as on which month Henry V began to be broadcast and when exactly the hypothesis was written.
	Spelling	examples with spelling errors.

Table 2: Summary of the Annotation Scheme. Toy examples are provided, ⇒ denotes entailment, ⇏ denotes contradiction. Only top and second level tags are provided, due to space considerations.

lored to the particular task at hand. Here, we balance these considerations and develop a novel NLI annotation scheme. We hope other large NLI datasets will be annotated according to our scheme to make even wider comparison possible.

Researchers have proposed many ways to ‘crack open the black box’ (Alishahi et al., 2019; Linzen et al., 2019), from uncovering lexical confounders or annotation “artifacts” (Gururangan et al., 2018; Geiger et al., 2018; Poliak et al., 2018b; Tsuchiya, 2018; Glockner et al., 2018; Geva et al., 2019) to evaluating generalization with diagnostic datasets (McCoy et al., 2019; Naik et al., 2018; Nie et al., 2019; Yanaka et al., 2019; Warstadt et al., 2019a; Geiger et al., 2020; Hossain et al., 2020; Jeretic et al., 2020; Warstadt et al., 2020; Schuster et al., 2020); see Zhou et al. (2020) for a critical overview. Specific to NLI, some have probed models to see what they learn (Richardson et al., 2019; Sinha et al., 2021b), honed data collection methods (Bowman et al., 2020; Vania et al., 2020; Parrish et al., 2021) and analyzed inherent disagreements between human annotators (Pavlick and Kwiatkowski, 2019; Nie et al., 2020b; Nangia et al., 2021), all in the service of understanding and improving models (see Poliak (2020) for a recent survey). See Table 10 and §A.3 for comparisons between our annotation scheme and others.

To inventory possible inference types, three NLP researchers independently inspected data from ANLI A1. For consistency, we then discussed and merged codes, applying an inductive approach (Thomas, 2006; Blodgett et al., 2021). Our scheme—provided in abbreviated form in Table 2—has 40 tag types that can be combined to a depth of up to four (see the Appendix for more details in §A.1, and more examples in Table 14). The top level of the scheme was fixed by the original ANLI paper to five classes: NUMERICAL, BASIC, REFERENCE, TRICKY inferences, and REASONING.[5] We aimed to balance proliferating narrow tags (and potentially being overly expressive), and limiting tag count to enable generalization (potentially being not expressive enough). A hierarchical tagset achieves the best of both worlds—we can measure all our metrics at varying granularities while allowing for pairs to receive as many tags as are warranted (see Table 1).

Annotation. Annotating NLI data for inference types requires various kinds of expert knowledge,

[5] These top-level types were introduced for smaller subsets of the ANLI development set in § 5 of Nie et al. (2020a), which we drastically expand both in number and specificity of tag types, as well as in annotation scope.

Dataset	Subset	Numerical	Basic	Reference	Tricky	Reasoning	Error
A1	All	40.8	31.4	24.5	29.5	58.4	3.3
	C	18.6	8.2	7.8	13.7	11.9	0.7
	N	7.0	9.8	7.1	6.4	31.3	1.0
	E	15.2	13.4	9.6	9.4	15.2	1.6
A2	All	38.5	41.2	29.4	29.1	62.7	2.5
	C	15.6	11.8	10.2	13.6	15.5	0.3
	N	8.1	12.8	9.1	7.4	30.0	1.4
	E	14.8	16.6	10.1	8.1	17.2	0.8
A3	All	20.3	50.2	27.5	25.6	63.9	2.2
	C	8.7	17.2	8.6	12.7	14.9	0.3
	N	4.9	13.1	8.2	4.6	30.1	1.0
	E	6.7	19.9	10.7	8.3	18.9	0.8

Table 3: Percentages (of the total) of tags by gold label and subdataset. 'All' refers to the total percentage of examples in that round that were annotated with that tag. 'C', 'N', and 'E', refer to percentage of examples with that tag that receive each gold label.

i.e. with a range of complicated linguistic phenomena and the particularities of the NLI task. Our work is fairly unique in that examples are *only* tagged as belonging to a particular branch of the taxonomy when the annotator believed the tagged phenomenon is required for a human to arrive at the target label assignment. Mere presence of a phenomenon was insufficient, meaning that automation was impossible, and expert annotation was necessary.[6] A single annotator with a decade's training in linguistics and expertise in NLI both devised our scheme and applied it to the ANLI development set. Annotation was laborious, taking the expert several hundred hours.

Inter-annotator Agreement. Employing a single annotator may have downsides, if they inadvertently introduce personal idiosyncrasies into their annotations. NLI may be especially susceptible to this, as recent work uncovers much variation in human judgements for this task (Pavlick and Kwiatkowski, 2019; Min et al., 2020; Nie et al., 2020b). To understand whether our tags are individual to the main annotator, we employed a second expert (with 5 years of linguistic training) to re-annotate 300 random examples, 100 from each development set. Re-annotation took the second annotator approximately 35 hours (excluding training time). Further details on the scheme, guidelines, and process are in Appendix A.

We measure inter-annotator agreement for each tag independently. For each example, annotators agree on a tag if they both used that tag or both did not use that tag; otherwise they disagree. Average percent agreement between our annotators is 72% for top-level and 91% for low-level tags respectively (see Table 8 and §A.2 for further details). Recall that 50% would be chance (since we are measuring whether the tag was used or not between two annotators). Our inter-annotator agreement is comparable to a similar semantic annotation effort on top of the original RTE data (Toledo et al., 2012), suggesting we have reached an acceptable level of agreement for our setting, and that the main annotator is not very idiosyncratic.

4 Experiments

We investigate 8 models: the original ANLI target model adversaries[7], and five SOTA models[8]—a RoBERTa-Large (Liu et al., 2019), a BART-Large (Lewis et al., 2019), an XLNet-Large (Yang et al. 2019, an ELECTRA-Large (Clark et al., 2020), and an ALBERT-XXLarge (Lan et al., 2020)—finetuned on SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), FEVER (Thorne et al., 2018), and ANLI rounds 1–3.

We report the tag distribution of the ANLI validation sets to establish an estimate of inference type frequency and explore what models may have learned as rounds progressed. To measure difficulty, we report models' correct label probability,

[6] Experts are well known to achieve higher performance than naïve crowdworkers when the task is linguistically complex (e.g., the CoLA subtask of the GLUE benchmark from Warstadt et al. (2019b), as well as Nangia and Bowman 2019, p. 4569, Basile et al. 2012; Bos et al. 2017, i.a.).

[7] For A2 and A3, which were ensembles, we randomly select a single RoBERTa-Large as the representative.

[8] https://github.com/facebookresearch/anli

and entropy on example pairs requiring each inference type (as accuracy on ANLI is still very low).

4.1 Tag Distribution

REASONING tags are the most common in the validation dataset, followed by NUMERICAL, TRICKY, BASIC, REFERENCE and then IMPERFECTIONS. The frequency of top-level tags are presented in Table 3, and for subtags in Table 15.

Walking through top-level types in turn, we find that NUMERICAL pairs are most common in A1. Since A1 contexts comprised the first few lines of Wikipedia entries—which often have numbers in them—this makes sense. A2, despite also using Wikipedia contexts, has a lower percentage of NUMERICAL examples, possibly because its target model—also trained on A1—improved on that category. In A3, the percentage of NUMERICAL pairs has dropped even lower. Between A1/A2 and A3, this drop in top level NUMERICAL tag frequency is due at least in part to a drop in the use of CARDINAL subtag, which results in a corresponding drop of third level DATES and AGES tags (in the Appendix). Overall, NUMERICAL pairs are more likely to have the gold label contradiction or entailment than neutral.

BASIC pairs are fairly common, with increasing frequency as rounds progress. Subtags LEXICAL and NEGATION rise sharply in frequency between A1 and A3; IMPLICATIONS and IDIOMS also rise—though they rise less sharply and are only present in $< 10\%$ of examples. COORDINATION and COMPARATIVES & SUPERLATIVES tag frequency stays roughly constant. Overall, BASIC examples tend to be gold labeled as entailment.

REFERENCE tags are rarest main tag type (present in 24.5% of A1 examples, rising slightly in A2 and A3). The most common subtag for REFERENCE is COREFERENCE with incidences ranging from roughly 16% in A1 to 26% in A3. Subtags NAMES and FAMILY maintain roughly constant low frequency across rounds, although there is a precipitous drop in NAMES tags for A3 (likely reflecting genre differences). Examples tagged as REFERENCE most commonly have entailment as their gold label for all rounds.

TRICKY inference types occur at relatively constant rates. A1 contains more examples with word reorderings than the others. PRAGMATIC examples are more prevalent in A1 and A3. A2 is unique in having slightly higher frequency of EXHAUSTIFICATION tags. WORDPLAY examples increase in A2 and A3 compared to A1. TRANSLATION pairs are rare (≈3%). On the whole, there are fewer neutral TRICKY pairs than contradictions or entailments, with contradiction being somewhat more common.

REASONING examples are very common across the rounds, with 50–60% of pairs receiving at least one. Subtagged FACTS pairs are also common, rising from 19% in A1 to roughly 25% of A2 and A3. CONTAINMENT shows the opposite pattern; it halves its frequency between A1 and A3. The frequency of third level LIKELY examples remains roughly constant whereas third level UNLIKELY and DEBATABLE examples become more common over the rounds. DEBATABLE tags rise to 3 times their rate in A1 by the third round, in part reflecting the contribution of different domains of text (see Table 7 for incidence on the procedural genre). On average, REASONING tags are more common for examples with a neutral gold label.

IMPERFECTION tags are rare across rounds ($\approx$ 14% of example pairs receive that tag on average), and are slightly more common for neutral pairs. SPELLING imperfections are the most common second level tag type, at $\approx 5-6\%$ of examples. Examples marked as AMBIGUOUS and ERROR were rare at $\approx 3-5\%$.

4.2 Model Predictions by Tag

For each model-round-tag triple, we report (i) the average probability of the correct prediction and (ii) the entropy of model predictions (i.e., from the input to the softmax layer) in Table 4[9]. We report both because neither number is fully interpretable in itself. Measuring the probability mass the model assigned to the correct label gives a nuanced notion of accuracy, whereas entropy can be seen as a measure of difficulty, in the sense that it can tell us how (un)certain a model is in its predictions. If a particular model-round-tag triple has high entropy, then that tag was more difficult for that model to learn from that round's data. A given model-round-tag triple can have both high probability and high entropy, which would show that the round-tag pairing is difficult (given the entropy), but that the model succeeded, at least to some extent, in learning how to predict the correct label anyway (given the probability).

ALBERT-XXLarge performs best overall, with

[9] Metrics for the lower level tags in Table 16–Table 20.

Round	Model	Numerical	Basic	Ref. & Names	Tricky	Reasoning	Imperfections
A1	BERT (R1)	0.10 (0.57)	0.13 (0.60)	0.11 (0.56)	0.10 (0.56)	0.12 (0.59)	0.13 (0.57)
	RoBERTa Ensemble (R2)	0.68 (0.13)	0.67 (0.13)	0.69 (0.15)	0.60 (0.18)	0.66 (0.15)	0.61 (0.14)
	RoBERTa Ensemble (R3)	0.72 (**0.07**)	0.73 (**0.08**)	0.72 (0.08)	0.65 (0.09)	0.70 (**0.08**)	0.68 (**0.07**)
	RoBERTa-Large	0.73 (0.13)	0.75 (0.12)	**0.76** (0.10)	**0.70** (0.14)	0.75 (0.15)	0.68 (0.13)
	BART-Large	0.73 (0.10)	0.76 (**0.08**)	0.72 (**0.07**)	**0.70** (**0.08**)	0.70 (0.11)	**0.71** (0.08)
	XLNet-Large	0.73 (0.10)	0.74 (0.09)	0.75 (0.09)	**0.70** (0.10)	0.72 (0.09)	0.67 (0.08)
	ELECTRA-Large	0.71 (0.29)	0.66 (0.36)	0.68 (0.34)	0.62 (0.44)	0.63 (0.41)	0.63 (0.40)
	ALBERT-XXLarge	**0.74** (0.22)	**0.77** (0.18)	**0.76** (0.20)	0.65 (0.21)	**0.77** (0.18)	0.69 (0.22)
A2	BERT (R1)	0.29 (0.53)	0.30 (0.47)	0.29 (0.44)	0.25 (0.48)	0.31 (0.47)	0.33 (0.48)
	RoBERTa Ensemble (R2)	0.19 (0.28)	0.21 (0.26)	0.20 (0.25)	0.16 (0.23)	0.19 (0.24)	0.19 (0.27)
	RoBERTa Ensemble (R3)	0.50 (0.18)	0.43 (0.16)	0.41 (0.14)	0.44 (0.14)	0.45 (0.14)	0.33 (0.14)
	RoBERTa-Large	0.54 (0.22)	0.51 (0.21)	0.47 (0.17)	0.48 (0.22)	0.49 (0.20)	0.49 (0.19)
	BART-Large	0.55 (0.13)	0.52 (0.13)	0.48 (0.14)	0.48 (0.15)	0.50 (0.13)	0.42 (**0.10**)
	XLNet-Large	0.54 (**0.11**)	0.53 (**0.12**)	0.53 (**0.13**)	**0.52** (**0.12**)	0.50 (**0.10**)	0.44 (**0.10**)
	ELECTRA-Large	0.56 (0.36)	0.53 (0.40)	0.52 (0.40)	0.51 (0.45)	0.53 (0.38)	0.54 (0.39)
	ALBERT-XXLarge	**0.57** (0.28)	**0.57** (0.29)	**0.58** (0.28)	0.50 (0.26)	**0.56** (0.25)	**0.58** (0.32)
A3	BERT (R1)	0.34 (0.53)	0.34 (0.51)	0.32 (0.50)	0.29 (0.55)	0.32 (0.49)	0.31 (0.54)
	RoBERTa Ensemble (R2)	0.29 (0.47)	0.26 (0.54)	0.26 (0.57)	0.24 (0.58)	0.27 (0.55)	0.23 (0.58)
	RoBERTa Ensemble (R3)	0.20 (0.43)	0.23 (0.50)	0.24 (0.53)	0.25 (0.54)	0.25 (0.54)	0.23 (0.52)
	RoBERTa-Large	0.44 (0.32)	0.44 (0.26)	0.45 (0.25)	0.49 (0.25)	0.46 (0.27)	0.40 (0.23)
	BART-Large	0.51 (**0.14**)	0.50 (**0.14**)	0.49 (**0.14**)	**0.53** (0.18)	0.50 (**0.14**)	0.48 (0.17)
	XLNet-Large	0.52 (0.15)	0.49 (**0.14**)	0.49 (0.15)	0.51 (**0.14**)	0.52 (0.15)	0.43 (**0.14**)
	ELECTRA-Large	0.55 (0.46)	0.51 (0.45)	0.52 (0.44)	0.54 (0.44)	0.52 (0.48)	0.47 (0.49)
	ALBERT-XXLarge	**0.56** (0.39)	**0.57** (0.33)	**0.55** (0.36)	0.52 (0.32)	**0.54** (0.32)	**0.52** (0.33)
ANLI	BERT (R1)	0.22 (0.54)	0.26 (0.52)	0.26 (0.50)	0.21 (0.53)	0.26 (0.51)	0.27 (0.53)
	RoBERTa Ensemble (R2)	0.41 (0.26)	0.37 (0.33)	0.34 (0.37)	0.33 (0.34)	0.35 (0.33)	0.32 (0.37)
	RoBERTa Ensemble (R3)	0.52 (0.20)	0.44 (0.27)	0.41 (0.30)	0.45 (0.26)	0.45 (0.28)	0.39 (0.28)
	RoBERTa-Large	0.59 (0.21)	0.55 (0.20)	0.53 (0.19)	0.56 (0.20)	0.56 (0.21)	0.50 (0.19)
	BART-Large	0.61 (**0.12**)	0.58 (**0.12**)	0.54 (**0.13**)	**0.57** (0.14)	0.55 (0.13)	0.52 (0.12)
	XLNet-Large	0.61 (**0.12**)	0.58 (**0.12**)	0.56 (**0.13**)	**0.57** (**0.12**)	0.57 (**0.12**)	0.50 (**0.11**)
	ELECTRA-Large	0.62 (0.35)	0.56 (0.40)	0.56 (0.40)	0.56 (0.44)	0.55 (0.43)	0.54 (0.44)
	ALBERT-XXLarge	**0.64** (0.28)	**0.63** (0.27)	**0.61** (0.30)	0.56 (0.26)	**0.61** (0.25)	**0.59** (0.30)

Table 4: Mean correct label probability (highest bold) and mean entropy of label predictions (lowest bold) by model and top level tag. Recall that the entropy for three equiprobable outcomes (i.e., random chance of three NLI labels) is upper bounded by ≈ 1.58. See Appendix E: Table 16–Table 21 for full results on lower-level tags.

the highest label probability for the full ANLI development set for each top-level tag except for Tricky, where it performs roughly as well as the others. BART-Large, XLNet-Large, and ELECTRA-Large are tied for second place, with RoBERTa-Large being a relatively close third. In general, the five SOTA models' probabilities of correct label differ by a few points, although BART-Large and XLNet-Large show markedly more certainty (i.e., lower entropy of predictions) than the others. It is clear that A1 is easier than A2 and A3, as measured by both higher correct label probability and lower entropy in general across models. A2 and A3 don't appreciably differ, although A3 generally has slightly lower correct label probabilities and higher entropies, meaning that A2 and A3 remain difficult for current models.

The ANLI model adversaries perform much worse that the SOTA models, having both lower mean probability of the correct label and often higher entropy: On A1 and A2, of three model adversaries, RoBERTa-Large (R3) also has the highest average label probability and lowest entropy (recall that RoBERTa-Large (R3) was one of the model adversaries in the ensemble, so its average prediction probability on A3 should be low).

Difficulty by Tag. Accuracy on ANLI is still fairly low (see Table 13), however it is still worth discussing which inference types confound our best current models. To understand our results, we have to be aware of how prevalent in the training corpus certain types are. We cannot necessarily expect a model to perform well on things it hasn't seen (although people often do, see Chomsky 1980). Because the ANLI training sets are not annotated, we will estimate the incidence of tags using the development sets (recall Table 3). To explore the relationship between phenomenon frequency and learnability by models, we split lower level tags into "common" tags are present in approximately 10% or more the ANLI development sets, while the rest are deemed "uncommon" (see Appendix E Table 16–Table 21 for more details).

Wikipedia	Fiction	News	Procedural	Legal	RTE
0.64 (**0.24**)	0.57 (0.29)	0.58 (**0.24**)	0.60 (0.28)	0.55 (0.39)	0.52 (0.52)

Table 5: Mean correct label probability (mean entropy of label predictions) for ALBERT-XXLarge by genre.

Tag	Wikipedia	Fiction	News	Procedural
Numerical	39.7%	3.5%	17.2%	10.5%
Basic	36.8%	41.0%	54.5%	48.5%
Reference	27.5%	21.0%	19.7%	14.5%
Tricky	29.0%	28.5%	25.3%	24.0%
Reasoning	61.7%	67.5%	59.6%	62.5%
Error	2.8%	3.5%	1.0%	2.5%

Table 6: Percentage of top-level tags in each genre.

Tag	Wikipedia	Fiction	News	Procedural
Numerical	**0.65** (0.25)	**0.65** (0.27)	**0.67** (0.32)	0.66 (**0.26**)
Basic	0.64 (0.25)	0.55 (0.27)	0.56 (0.22)	0.61 (0.32)
Reference	**0.65** (**0.22**)	0.50 (**0.23**)	0.52 (0.23)	**0.71** (0.29)
Tricky	0.56 (0.24)	0.52 (0.26)	0.64 (**0.19**)	0.57 (0.29)
Reasoning	0.66 (0.24)	0.61 (0.28)	0.55 (0.24)	0.56 (0.31)
Imperfection	0.63 (0.27)	0.62 (0.34)	0.60 (0.26)	0.53 (0.26)

Table 7: Mean correct label probability (mean entropy of label predictions) for ALBERT-XXLarge.

Perhaps obviously, common inference types (e.g., Reasoning-Lexical, Numerical-Dates, Reasoning-Likely) are easier for models to perform well on (according to higher correct label probability). More compellingly though, there were some common inference types that the models still behaved poorly on, namely Reasoning-Facts, Reference-Coreference, Basic-Negation and Tricky-Syntactic. Since these tags are fairly frequent, it's reasonable to conclude that these types required more complex knowledge. For example, Reasoning-Facts, which includes knowing that "2020 is this year" or that "a software engineering tool can't enable people to fly".

Models can do fairly well on some uncommon tags, e.g., Basic-Coordination and Numerical-Nominal, Reasoning-Unlikely, Reference-Names, Reasoning-Containment, Tricky-wordplay. There are two potential explanations for this higher than expected performance: perhaps the SNLI, MNLI or FEVER training data has sufficient quantities these inference types or, alternatively, these types are somewhat easier to learn from fewer examples. Models do struggle with Numerical-Counting, Numerical-Age, Basic-Implications, Reasoning-Debatable, Basic-Idiom, Tricky-Pragmatics, Tricky-Exhaustification. Similarly, these failures can either be due to tag rarity or to their inherent difficulty. Future work could ask whether augmenting training data with more examples of these types boosts performance.

Overall, models struggle with examples requiring linguistic or external knowledge: the hardest top-level tag for all models is Tricky, with Reasoning and Reference being next in line. Anywhere from one quarter to two thirds of data contains at least one of these tags, so models have been exposed to these inference types. Numerical and Basic examples are less difficult, but are by no means solved. On rounds A1–3, adversaries improve on Numerical examples, suggesting that exposure to relevant Numerical examples can enable modest improvement (see also Dua et al. 2019 for a related observation).

Summary. ALBERT-XXLarge performs slightly better than the others, but it is less certain in its predictions; XLNet-Large and BART-Large perform slightly worse, but have lower entropy. Top-level Tricky[10], Reasoning, and Reference categories are still difficult for SOTA models, even though they are frequent. Of the lower level tags that appear in approximately 10% of the ANLI development sets, Facts, Coreference, Negation and Syntactic example pairs remain difficult.

4.3 Overlap in Model Predictions

Generally, model outputs were somewhat correlated with ANLI gold labels represented as one-hot vectors (see Figure 1). ALBERT-XXLarge model outputs are the most positively correlated (Pearson's correlation) (≈ 0.5), RoBERTa-Large, BART-Large, XLNet-Large, and ELECTRA-Large have medium sized positive correlations, and the R2 and R3 RoBERTa-Large models have small positive correlations. BERT (R1) is slightly negatively correlated with gold labels. All differences were significant ($p < 0.01$).

However, different models made very similar predictions: RoBERTa-Large, BART-Large, XLNet-Large, and ALBERT-XXLarge correlated highly with each other (> 0.6), with ELECTRA-Large (> 0.5), and with A2 and A3 RoBERTa-

[10] Tricky was the only inference type for which ALBERT-XXLarge wasn't the top performer; XLNet-Large performed somewhat better, largely due to stronger higher probability and lower entropy on linguistically sophisticated Syntactic and Pragmatic examples.

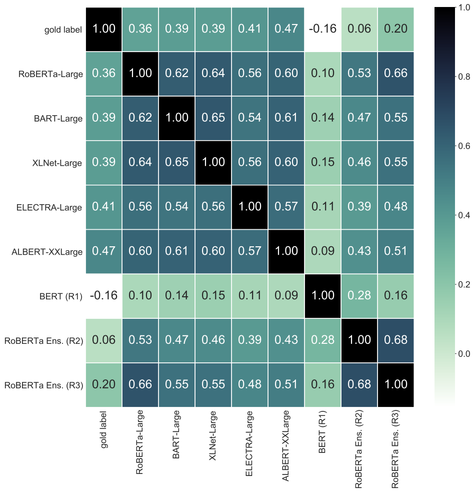


Figure 1: Correlation between gold labels and model outputs. All comparisons are significant $p < 0.01$.

Large models $(0.4 - 0.5)$. RoBERTa-Large model predictions from A2 correlated with those from A3 (0.68). These results suggest that substantial improvement on ANLI may require radically new ideas, not just minor adjustments to the pretrain-finetune paradigm (c.f. Sinha et al. 2021a,b).

4.4 Analyzing Results by Genre

A3 was collected using contexts from a variety of text domains. Table 5 shows the performance of the highest performing model (ALBERT-XXLarge) across genres. Wikipedia is the least difficult genre (as well as the most frequent), Procedural is somewhat harder, then News (which is lower entropy), followed by Fiction, Legal, then RTE. Genres differ widely in how many of their examples have particular top-level tags (see Table 6). Across all genres, TRICKY and REASONING examples occur at roughly the same rates—with REASONING examples being very common across the board. Compared to the other genres, News text has more BASIC tags, and Wikipedia text has more NUMERICAL. Procedural text has the lowest rate of NUMERICAL and REFERENCE tags, but the highest rate of IMPERFECTION.

Table 7 breaks down of the performance of the ALBERT-XXLarge model by genre and tag (see Table 22 in the Appendix for the other models' performance). ALBERT-XXLarge performance on NUMERICAL examples is relatively stable across the genres, but for the other top level tags there is some variation that does not just reflect tag frequency. For example, the ALBERT-XXLarge model does better on BASIC and REASONING examples from Wikipedia, on REFERENCE examples from the Procedural genre, and on TRICKY examples from the News genre. This suggests that data from different genres could be differentially beneficial for training the skills needed for these top-level tags, suggesting that targeted upsampling could be beneficial in the future.

4.5 Other Analyses

Appendix B provides a detailed analysis of other dataset properties (word and sentence length, and most common words by round, gold label, and tag), where we show that ANLI and MNLI are relatively similar to each other but differ from SNLI. Crowdworker rationales from ANLI are explored in §B.1, Table 23–Table 24.

5 Conclusion

We release annotations of the ANLI development sets to determine which inference types are responsible for model success and failure, and how their frequencies change over dynamic data collection. Inferences relying on numerical or common sense reasoning are most prevalent, appearing in ≈40%–60% of examples. We finetuned a variety of transformer language models on NLI and compared their performance to the original target models used to adversarially collect ANLI. ALBERT-XXLarge performs the best of our 8 model sample, but there is still ample room for improvement in accuracy. Despite being frequent, examples requiring common sense reasoning, understanding of co-reference, negation and syntactic knowledge remain the most difficult. One could imagine explicit interventions to address this, perhaps incorporating insights from Sap et al. (2020), or using other modes of evaluation that explore model and data dynamics (Gardner et al., 2020; Swayamdipta et al., 2020; Rodriguez et al., 2021).

ANLI remains difficult: the huge GPT-3 model (Brown et al., 2020) barely made any progress, and even the recent DeBERTa model (He et al., 2021) cannot break 70% accuracy. We hope our annotations will inspire new innovations by enabling more fine-grained understanding of model strengths and weaknesses as ANLI matures.

Acknowledgements

Special thanks to Naman Goyal for converting the R2 and R3 RoBERTa models to something modern `fairseq` can load. Thanks as well to Yixin Nie, Mohit Bansal, Emily Dinan, Grusha Prasad, Nikita Nangia, and Alex Wang for comments and suggestions along the way. Thanks as well to several rounds of anonymous reviewers from the *ACL community and SCIL.

References

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. *Natural Language Engineering*, 25(4):543–557.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3196–3200, Istanbul, Turkey. European Languages Resources Association (ELRA).

Isaac I Bejar, Roger Chaffin, and Susan Embretson. 2012. *Cognitive and psychometric analysis of analogical problem solving*. Springer Science & Business Media.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2019. What kind of natural language inference are nlp systems learning: Is this enough? In *International Conference on Agents and Artificial Intelligence (ICAART2)*, pages 919–931.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. Stereotyping Norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. 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 1004–1015, Online. Association for Computational Linguistics.

Johan Bos, Valerio Basile, Kilian Evang, Noortje J Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In *Handbook of linguistic annotation*, pages 463–496. Springer.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Samuel R. Bowman, Jennimaria Palomaki, Livio Baldini Soares, and Emily Pitler. 2020. New protocols and negative results for textual entailment data collection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8203–8214, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2020. Uncertain natural language inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8772–8779, Online. Association for Computational Linguistics.

Gennaro Chierchia et al. 2004. Scalar implicatures, polarity phenomena, and the syntax/pragmatics interface. *Structures and beyond*, 3:39–103.

Noam Chomsky. 1980. *On cognitive structures and their development: A reply to Piaget*. Harvard University 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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Peter Clark. 2018. What knowledge is needed to solve the RTE5 textual entailment challenge?

Robin Cooper, Crouch Dick, Jan van Eijck, Chris Fox, Joseph van Genabith, Han Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Brisco, Holger Maier, and Karsten Konrad. 1996. Using the framework. technical report lre 62-051r. The FraCaS consortium.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

DataPerf Working Group. 2021. DataPerf: Benchmarking data for better ML. Technical report.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

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.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Yaroslav Fyodorov, Yoad Winter, and Nissim Francez. 2000. A natural logic inference system. In *Proceedings of the 2nd Workshop on Inference in Computational Semantics (ICoS-2)*. Citeseer.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. 2018. Stress-testing neural models of natural language inference with multiply-quantified sentences. *arXiv preprint arXiv:1810.13033*.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1161–1166. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

H Paul Grice. 1975. Logic and conversation. *1975*, pages 41–58.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. 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 107–112. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: decoding-enhanced BERT with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Paloma Jeretic, 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.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Pratik Joshi, Somak Aditya, Aalok Sathe, and Monojit Choudhury. 2020. TaxiNLI: Taking a ride up the

NLU hill. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 41–55, Online. Association for Computational Linguistics.

Alain Joubert, Mathieu Lafourcade, and Nathalie Le Brun. 2018. The jeuxdemots project is 10 years old: What we have learned. *Games and Gamification for Natural Language Processing*, 22.

David Jurgens, Saif Mohammad, Peter Turney, and Keith Holyoak. 2012. SemEval-2012 task 2: Measuring degrees of relational similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 356–364, Montréal, Canada. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, Dieuwke Hupkes, and et al., editors. 2019. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Florence, Italy.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Peter LoBue and Alexander Yates. 2011. Types of common-sense knowledge needed for recognizing textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 329–334, Portland, Oregon, USA. Association for Computational Linguistics.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. *Advances in Neural Information Processing Systems*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nikita Nangia and Samuel R. Bowman. 2019. Human vs. muppet: A conservative estimate of human performance on the GLUE benchmark. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4566–4575, Florence, Italy. Association for Computational Linguistics.

Nikita Nangia, Saku Sugawara, Harsh Trivedi, Alex Warstadt, Clara Vania, and Samuel R. Bowman. 2021. What ingredients make for an effective crowdsourcing protocol for difficult NLU data collection tasks? 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 1221–1235, Online. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of NLI models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6867–6874.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020a. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020b. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alex Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R Bowman. 2021. Does putting a linguist in the loop improve nlu data collection? *arXiv preprint arXiv:2104.07179*.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Trans. Assoc. Comput. Linguistics*, 7:677–694.

Adam Poliak. 2020. A survey on recognizing textual entailment as an NLP evaluation. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 92–109, Online. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018a. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018b. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Kyle Richardson, Hai Hu, Lawrence S Moss, and Ashish Sabharwal. 2019. Probing natural language inference models through semantic fragments. *arXiv preprint arXiv:1909.07521*.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? 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 4486–4503, Online. Association for Computational Linguistics.

Ohad Rozen, Vered Shwartz, Roee Aharoni, and Ido Dagan. 2019. Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675, Online. Association for Computational Linguistics.

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. ConjNLI: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online. Association for Computational Linguistics.

Mark Sammons, V.G.Vinod Vydiswaran, and Dan Roth. 2010. "ask not what textual entailment can do for you...". In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1199–1208, Uppsala, Sweden. Association for Computational Linguistics.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33, Online. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Colorado.

Sebastian Schuster, Yuxing Chen, and Judith Degen. 2020. Harnessing the linguistic signal to predict scalar inferences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5387–5403, Online. Association for Computational Linguistics.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021a. 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.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021b. UnNatural Language Inference. 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 7329–7346, Online. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

David R Thomas. 2006. A general inductive approach for analyzing qualitative evaluation data. *American journal of evaluation*, 27(2):237–246.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Assaf Toledo, Sophia Katrenko, Stavroula Alexandropoulou, Heidi Klockmann, Asher Stern, Ido Dagan, and Yoad Winter. 2012. Semantic annotation for textual entailment recognition. In *Mexican International Conference on Artificial Intelligence*, pages 12–25. Springer.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of LREC*.

Clara Vania, Ruijie Chen, and Samuel R. Bowman. 2020. Asking Crowdworkers to Write Entailment Examples: The Best of Bad options. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 672–686, Suzhou, China. Association for Computational Linguistics.

Eric Wallace, Adina Williams, Robin Jia, and Douwe Kiela. 2021. Analyzing dynamic adversarial training data in the limit. *arXiv preprint arXiv:2110.08514*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019a. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019b. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. *International Conference on Learning Representations*.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Aaron Steven White and Kyle Rawlins. 2018. The role of veridicality and factivity in clause selection. In *Proceedings of the 48th Annual Meeting of the North East Linguistic Society*.

Aaron Steven White, Elias Stengel-Eskin, Siddharth Vashishtha, Venkata Subrahmanyan Govindarajan, Dee Ann Reisinger, Tim Vieira, Keisuke Sakaguchi, Sheng Zhang, Francis Ferraro, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2020. The universal decompositional semantics dataset and decomp toolkit. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5698–5707, Marseille, France. European Language Resources Association.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 250–255, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online. Association for Computational Linguistics.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.

A Further Details on Annotation

A.1 Details of the Annotation Scheme

A full ontology, comprising all four levels, is provided together with examples in Table 14.

To give an idea of what sorts of information falls under each tag, we will go through them in turn. Numerical classes refer to examples where numerical reasoning is crucial for determining the correct label, and break down into Cardinal, Ordinal—along the lines of Ravichander et al. (2019)—Counting and Nominal; the first two break down further into Ages and Dates if they contain information about either of these topics. Basic consists of staple types of reasoning, such as lexical hyponymy and hypernymy (see also Glockner et al. 2018), conjunction (see also Toledo et al. 2012; Saha et al. 2020), and negation (see also Hossain et al. 2020). Reference consists of pairs that require noun or event references to be resolved (either within or between context and hypothesis). Tricky examples require either complex linguistic knowledge, say of pragmatics or syntactic verb argument structure, reorderings, word games (e.g., anagrams, acrostic jokes), and foreign language content (Translation).

Reasoning examples require the application of reasoning outside of what is provided in the example alone; it is divided into three levels. The first is Plausibility, which was loosely inspired by Bhagavatula et al. (2020); Chen et al. (2020), for which the annotator provided their subjective intuition on how likely the situation is to have genuinely occurred (for example 'when computer games come out they are often buggy' and 'lead actors get paid the most' are likely). Plausibility also contains Debatable examples, which depend on opinion or scalar adjectives like "big" (e.g. a big mouse is "big" for a mouse, but not big when compared to an elephant). The other two Facts and Containment refer to external facts about the world (e.g., 'what year is it now?') and relationships between things (e.g., 'Australia is in the southern hemisphere'), respectively, that were not clearly provided by the example pair itself.

There is also a catch-all class labeled Imperfection that catches not only label "errors" (i.e., rare cases of labels for which the expert annotator(s) disagreed with the gold label from the crowdworker-annotator), but also spelling mistakes (Spelling), event coreference examples (EventCoref[11]), and pairs that could reasonably be given multiple correct labels (Ambiguous). The latter are likely uniquely subject to human variation in entailment labels, *à la* Pavlick and Kwiatkowski (2019), Min et al. (2020), Nie et al. (2020b), since people might vary on which label they initially prefer, even though multiple labels might be possible.

Exhaustive List of Tags. In the actual dataset, tags at different levels are dash-separated, as in Reasoning-Plausibility-Likely. These include: Basic CauseEffect, Basic ComparativeSuperlative, Basic Coordination, Basic Facts, Basic Idioms, Basic Lexical Dissimilar, Basic Lexical Similar, Basic Modus, Basic Negation, EventCoref, Imperfection Ambiguity, Imperfection Error, Imperfection NonNative, Imperfection Spelling, Numerical Cardinal, Numerical Cardinal Age, Numerical Cardinal Counting, Numerical Cardinal Dates, Numerical Cardinal Nominal, Numerical Cardinal Nominal Age, Numerical Cardinal Nominal Dates, Numerical Ordinal Numerical Ordinal Age, Numerical Ordinal Dates, Numerical Ordinal Nominal, Numerical Ordinal Nominal Dates, Reasoning CauseEffect, Reasoning Containment Location, Reasoning Containment Parts, Reasoning Containment Times, Reasoning Debatable, Reasoning Facts, Reasoning-Plausibility Likely, Reasoning Plausibility Unlikely, Reference Coreference, Reference Family, Reference Names, Tricky Exhaustification, Tricky Pragmatic, Tricky Syntactic, Tricky Translation, Tricky Wordplay.

In addition to these tags, some top-level tags are associated with a -0 flag; these are very rare (less than 30 of these in the dataset). The zero-flag was associated with examples that didn't fall into any lower level categories. Finally, for the purposes of this paper, we collapsed two second-level tags Basic CauseEffect and Basic Modus[12] into Basic-Implications because these types were rare, we felt the two are related.

[11] SNLI and MNLI annotation guidelines required annotators to assume event coreference.

[12] Modus labeled classical inference types such as Modus Ponens, Modus Tollens, etc.

Tag	Agreement (%)	A1 # of Tags	A2 # of Tags
REASONING	59.1%	176	226
BASIC	69.2%	122	128
REFERENCE	64.5%	88	136
NUMERICAL	88.6%	94	112
TRICKY	64.5%	89	105
IMPERFECTION	81.2%	44	56
EVENTCOREF	89.2%	11	29
REASONING-FACTS	54.8%	61	174
REFERENCE-COREFERENCE	66.2%	72	109
REASONING-PLAUSIBILITY	71.2%	104	70
BASIC-LEXICAL	73.9%	67	69
NUMERICAL-CARDINAL-DATES	**92.9%**	51	68
TRICKY-PRESUPPOSITION	74.9%	19	66
BASIC-NEGATION	**94.3%**	34	33
REFERENCE-NAMES	82.2%	22	45
NUMERICAL-CARDINAL	**92.9%**	23	38
BASIC-CONJUNCTION	87.9%	12	38
TRICKY-SYNTACTIC	88.2%	33	12
EVENTCOREF	89.2%	11	29
TRICKY-TRANSLATION	**92.6%**	15	23
TRICKY-EXHAUSTIFICATION	**94.6%**	22	14
IMPERFECTION-SPELLING	**93.3%**	15	15
REASONING-CONTAINMENT-LOCATION	**96.6%**	15	13
NUMERICAL-CARDINAL-AGE	**98.6%**	14	12
IMPERFECTION-NONNATIVE	**94.3%**	5	20
IMPERFECTION-LABEL	**93.6%**	8	17
IMPERFECTION-AMBIGUITY	**93.9%**	16	8
BASIC-COMPARATIVESUPERLATIVE	**95.3%**	17	5
REASONING-CONTAINMENT-TIME	**94.3%**	16	5
BASIC-CAUSEEFFECT	**95.9%**	8	12
NUMERICAL-ORDINAL	**98.6%**	9	9
NUMERICAL-CARDINAL-COUNTING	**99.3%**	7	9
NUMERICAL-CARDINAL-NOMINAL-DATES	**95.3%**	0	14
TRICKY-WORDPLAY	**96.9%**	8	5
NUMERICAL-CARDINAL-NOMINAL	**96.3%**	6	7
BASIC-IDIOM	**96.3%**	7	6
REFERENCE-FAMILY	**99.3%**	5	5
NUMERICAL-ORDINAL-DATES	**97.9%**	4	2
BASIC-0	**98.3%**	4	1
IMPERFECTION-0	**98.9%**	2	1
REASONING-CONTAINMENT-PARTS	**99.6%**	1	0
REASONING-0	**99.6%**	0	1
Aggregate	**91.1% (avg)**	713 (sum)	955 (sum)

Table 8: Interannotator agreement percentages (bold exceeded 90%) and tag counts for 300 randomly sampled examples. Tags are sorted by the number of usages of that tag by either annotator.

More Examples from the Annotation Guidelines. Some tags required sophisticated linguistic domain knowledge, so more the annotation guidelines included more examples (some will be provided here). For example, the TRICKY-EXHAUSTIFICATION is wholly novel, i.e., not adopted from, or similar to, any other semantic annotation scheme known to the authors. This tag marks examples where the original crowdworker-annotator assumed that only one predicate holds of the topic, and that other predicates don't. Often TRICKY-EXHAUSTIFICATION examples have the word "only" in the hypothesis, but that's only a tendency: observe the context, *Linguistics is the scientific study of language, and involves an analysis of language form, language meaning, and language in context* and the hypothesis *Form and meaning are the only aspects of language linguistics is concerned with*, which gets labeled as a contradiction.[13] For this example, the crowdworker-annotator wrote a hypothesis that excludes one of the core properties of linguistics provided in the context and claims that the remaining two they list are the only core linguistic properties.

To take another example, also a contradiction: For the context, *The Sound and the Fury is an American drama film directed by James Franco. It is the second film version of the novel of the same name by William Faulkner* and hypothesis *Two Chainz actually wrote The Sound and the Fury*, we have a TRICKY-EXHAUSTIFICATION tag. The Gricean Maxims of Relation and Quantity (Grice, 1975) require the writer of the original context to be maximally cooperative and informative, and thus, to list all the authors of *The Sound and Fury*. Since the context only listed Faulkner, we con-

[13] This example also receives BASIC-COORDINATION, and BASIC-LEXICAL-SIMILAR for "involves" and "aspects"/"concerned with".

clude that the book only had one author, Faulkner, and Two Chainz did *not* in fact (co-)author *The Sound and the Fury*.[14]

As we mentioned above, any one example sentence pair can receive multiple tags. An example with a hypothesis *George III comes after George II* would receive tags Reference-Names (because we are comparing the names of two individuals), and Numerical-Ordinal (because we are comparing the roman numerals for first and second). A pair with the context *Sean Patrick Hannity...is an American talk show host, author, and conservative political commentator...* and the hypothesis *Hannity has dated a liberal* would receive the tags Basic-Lexical (because of the relation between "conservative" and "liberal"), Reference-Coreference, (because of the coreference between "Sean Patrick Hannity" and "Hannity"), and Reasoning-Unlikely (because it's unlikely given world knowledge that a liberal and a conservative commentator would date, although it's definitely possible).

The annotation guidelines also provided examples to aid in disentangling Reference-Names from Reference-Coreference, as they often appear together. Reference-Coreference should be used when resolving reference between non-string matched noun phrases (i.e. DPs) is necessary to get the label: ***Mary Smith***$_i$ *was a prolific author.* ***She***$_i$ *had a lot of published works by 2010.*⇒***Smith***$_i$ *published many works of literature.* Reference-Names is used when the label is predicated on either (i) a discussion of names, or (ii) resolving multiple names given to a person, but the reference in the hypothesis is an exact string match to one of the options: ***La Cygne***$_i$ *(pronounced **"luh SEEN"**) is a city in the south of France.*⇒***La Cygne***$_i$ *is in France.* Some examples require both Reference-Coreference and Reference-Names tags: ***Mary Beauregard Smith, the fourth grand Princess of Winchester*** *was a prolific author.*⇒***Princess Mary*** *wrote a lot.*

A.2 Inter-Annotator Agreement

Annotation guidelines for each tag were discussed verbally between the two annotators during the training of the second expert. The main expert annotator trained the second by first walking through the annotation guidelines (i.e., Table 2), answering any questions, and providing additional examples taken from their experience as necessary. The second expert then annotated 20 randomly sampled examples from the R1 training set as practice.

	Average			Top Level Tags		
	Precision	Recall	F1	Precision	Recall	F1
A_1	0.55	0.42	0.44	0.59	0.73	0.61
A_2	0.42	0.55		0.73	0.59	

Table 9: Average Precision, Recall and F1 between our two annotators on 300 randomly selected development set examples. A_1 was taken with the original annotator as ground truth, A_2 with the second expert. Recall that X to Y precision is equivalent to Y to X recall.

The two annotators subsequently discussed their selections on these training examples when they differed. Of course, there is some subjectivity inherent in this annotation scheme, which crucially relies on expert opinions about what information in the premise or hypothesis could be used to determine the correct label. After satisfactorily coming to a conclusion (i.e., a consensus for all 20 examples), the second annotator was provided with another set of 20 randomly sampled examples, this time from the R3 training set (to account for genre differences across rounds), and again, discussion was repeated until consensus was reached. Several further discussions took place. Once both annotators were confident in the second expert annotator's understanding of the scheme, the secondary annotator was provided with 3 random selections of 100 examples (one from each development set) as the final set to calculate inter-annotator agreement from. The second annotator was also provided with the exhaustive tag list (above), which includes some splits that subcategorize the tags from Table 2 even further. The tags are visible in Table 8, along with percent agreement for each tag.

To provide additional NLI-internal context for our percent agreement results, we note that percent agreement on both top and lower level tags exceeds the percent agreement of non-experts on the task of NLI as reported in Bowman et al. (2015) and Williams et al. (2018). Recall that performing NLI is a subtask of our annotations (i.e., experts must check the NLI label to determine if there was an error and must also then tag contained phenomena that contribute to the label decision).

Since our annotation scheme incorporated some subjectivity—i.e., annotators tag phenomena they

[14] This pair also gets Tricky-Pragmatic, and Event-Coref and Basic-Lexical-Similar tags.

Our Scheme's Tag	Other Scheme's Tag (Citation)
BASIC-NEGATION	Negation (Naik et al., 2018; Hossain et al., 2020; Geiger et al., 2020)
BASIC-LEXICAL-DISSIMILAR	Antonymy (Naik et al., 2018), Contrast (Bejar et al., 2012); Ch. 3[15]
BASIC-LEXICAL-SIMILAR	Overlap (Naik et al., 2018), Similar (Bejar et al., 2012); Ch. 3, hyponym/hypernym (Geiger et al., 2020), Lexical (Joshi et al., 2020)
BASIC-CAUSEEFFECT	Cause-Purpose (Bejar et al., 2012); Ch. 3, cause (Sammons et al., 2010), Cause and Effect (LoBue and Yates, 2011)
BASIC-COORDINATION	Conjoined Noun Phrases (Cooper et al., 1996), ConjNLI (Saha et al., 2020), "Connectives" (Joshi et al., 2020)
BASIC-COMPARATIVESUPERLATIVE	Comparatives (Cooper et al., 1996), , "Connectives" (Joshi et al., 2020)
NUMERICAL	numeric reasoning, numerical quantity (Sammons et al., 2010), Mathematical (Joshi et al., 2020)
NUMERICAL-CARDINAL	cardinal (Ravichander et al., 2019)
NUMERICAL-ORDINAL	ordinal (Ravichander et al., 2019)
REFERENCE-COREFERENCE	Anaphora (Inter-Sentential, Intra-Sentential) (Cooper et al., 1996), coreference (Sammons et al., 2010)
REFERENCE-COREFERENCE with REFERENCE-NAMES	Representation (Bejar et al., 2012); Ch. 3
REFERENCE-FAMILY	parent-sibling, kinship (Sammons et al., 2010)
REFERENCE-NAMES	name (Sammons et al., 2010)
REASONING-DEBATABLE	Cultural/Situational (LoBue and Yates, 2011), Defeasible Inferences (Rudinger et al., 2020)
REASONING-PLAUSIBILITY-LIKELY	Probabilistic Dependency (LoBue and Yates, 2011)
REASONING-CONTAINMENT-TIMES	Temporal Adverbials (Cooper et al., 1996), Space-Time (Bejar et al., 2012); Ch. 3, event chain, temporal (Sammons et al., 2010)
REASONING-CONTAINMENT-LOCATION	spatial reasoning (Sammons et al., 2010), Geometry (LoBue and Yates, 2011)
REASONING-CONTAINMENT-PARTS	Part-Whole, Class-Inclusions (Bejar et al., 2012); Ch. 3, has-parts (LoBue and Yates, 2011)
REASONING-FACTS	Real World Knowledge (Naik et al., 2018; Clark, 2018; Bernardy and Chatzikyriakidis, 2019)
TRICKY-SYNTACTIC	passive-active, missing argument, missing relation, simple rewrite, (Sammons et al., 2010)
IMPERFECTIONS-AMBIGUITY	Ambiguity (Naik et al., 2018)

Table 10: Comparisons between our tagset and tags from other annotation schemes.

believe a human would use to provide the NLI label for the example—annotators are likely to have different blindspots. Descriptively, annotators differed slightly in the number of tags they assign on average: the original annotator assigns fewer tags per example (Mean $= 2.25$, Std. $= 1.01$) than the second expert (Mean $= 3.02$, Std. $= 1.45$). The number of tags in the intersection of the two was predictably lower (Mean $= 1.20$, Std.$= 0.85$) than either annotator's average or the union (Mean $= 4.07$, Std. $= 1.55$).

In addition to agreement percentages that are reported in Table 8, we report average precision, recall, and F1 (a weighted average of the two) for our annotations in Table 9.[16] For percentages, we note that agreement was generally higher for rarer tags. The most frequent top-level tag, REASONING, had the lowest agreement, perhaps due to disagreements in REASONING-FACTS, where the subjectivity of decisions likely drove down agreement. Subjectivity might be expected for REASONING-PLAUSIBILITY examples as well, because it is hard to be sure whether a particular fact is necessary for the label (particular in the case of REASONING-PLAUSIBILITY-DEBATABLE. REASONING-PLAUSIBILITY also showed some disagreement, as people differ whether they feel compelled to note that the likelihood of a context is relevant for the label decision. Finally, we note that frequent lower level tags NUMERICAL-CARDINAL(-DATES) and BASIC-NEGATION had the highest agreement.

Although we report accuracy (i.e., percentage agreement), F1 is usually more useful than accuracy, especially if you have an uneven class distribution (as we do). For this reason, we additionally report F1, precision and recall between the two annotators (reporting statistics twice, once with each annotator taken to be ground truth). Precision, recall and F1 are all fairly high (recall that these three measures are upper bounded by 1), but are higher for top level tags than for the average of all tags. We believe this is an acceptable level of agreement, especially given the difficulty of the task and the fact that tags vary in how subjective their decisions are.

A.3 Direct Comparisons to other Annotation Schemes

Our scheme derives its inspiration from the wealth of prior work on types of sentential inference both within and from outside NLP—Cooper et al. (1996); Sammons et al. (2010); LoBue and Yates (2011); Jurgens et al. (2012); Jia and Liang (2017); White et al. (2017); Naik et al. (2018); Nie et al. (2019); Kim and Linzen (2020); Yu and Ettinger (2020); White et al. (2020), i.a. When one implements an annotation scheme, one must decide on the level of depth one wants to achieve. On the one hand, a small number of tags can allow for easy annotation (by non-experts or even automatically), whereas on the other, a more complicated and complete annotation scheme (like, e.g., Cooper et al. 1996; Bejar et al. 2012) can allow for a better understanding of the full range of possible phenomena that might be relevant. (Note: for contextualization, our tags are greater in tag number than Naik et al. (2018) but smaller and more manageable than Cooper et al. 1996 and Bejar et al. 2012). We wanted annotations that allow for an

[16] For all statistics that aggregate tag results, we did not include Imperfection tags, as imperfections can be difficult to spot and annotator differences for these tags typically only represent whether an annotator noticed a mistake when the other did not.

evaluation of model behavior on a phenomenon-by-phenomenon basis, in the spirit of Weston et al. (2016); Wang et al. (2018); Jeretic et al. (2020)—but unlike Jia and Liang (2017). We also wanted to be able to detect interactions between phenomena (Sammons et al., 2010). Thus, we implemented our hierarchical scheme (for flexible tag-set size) in a way that could provide all these desiderata.

Table 10 provides a by-tag comparison between our annotation scheme and several others. Only direct comparisons are listed in the table; in other cases, our scheme had two tags where another scheme had one, or vice versa. Some of these examples are listed below, by the particular inference types for each annotation scheme.

Several labels from Naik et al. (2018)'s annotation scheme concur with ours, but ours has much wider coverage. In fact, it is a near proper superset of their scheme. Both taxonomies have a NEGATION tag, an AMBIGUITY tag, a REAL WORLD KNOWLEDGE—which for us is REASONING-FACTS, and a ANONYMY tag—which for us is BASIC-LEXICAL-DISSIMILAR. Additionally, both annotation schemes have a tag for numerical reasoning. We didn't include "word overlap" as that is easily automatable and would thus be an inappropriate use of limited hand-annotation time. Instead, we include a more flexible/complex notion of overlap in our BASIC-LEXICAL-SIMILAR tag, which accounts not only for synonyms, but also for phrase level paraphrases.

Our scheme can handle nearly all of the inference types in Sammons et al. (2010). For example, their 'numerical reasoning' tag maps onto a combination of NUMERICAL tags and REASONING-FACTSfor us to account for external mathematical knowledge. A combination of their 'kinship' and 'parent-sibling' tags is present in our REFERENCE-FAMILY tag. One important difference between our approach and theirs is that we do not separate negative and positive occurrences of phenomena; both would appear under the same tag for us. One could imagine performing a further round of annotation on the ANLI data to separate positive from negative as Sammons et al. does.

Several of the intuitions of the LoBue and Yates (2011) taxonomy are present in our scheme. For example, their 'arithmetic' tag roughly corresponds to a combination of our NUMERICAL-CARDINAL and REASONING-FACTS (i.e., for mathematical reasoning). Examples labeled with their "preconditions" tag would receive our TRICKY-PRAGMATIC tag. Interestingly, our TRICKY-EXHAUSTIFICATION tag seems to be a combination of their 'mutual exclusivity', 'omniscience' and 'functionality' tags. Other relationships between our tags and theirs are in Table 10.

Many of our numerical reasoning types were inspired by Ravichander et al. (2019), which showed that many NLI systems perform very poorly on many types of numerical reasoning. In addition to including cardinal and ordinal tags, as they do, we take their ideas one step further and also tag numerical examples where the numbers are not merely playing canonical roles as degrees of measure (e.g., NUMERICAL-NOMINAL and NUMERICAL-COUNTING). We also expand on their basic numerical types by specifying whether a number refers to a date or an age. For any of their examples requiring numerical reasoning, we would assign NUMERICAL as a top level tag, as well as a REASONING-FACTS tag, as we described in the paragraph above. A similar set of tags would be present for their "lexical inference" examples where, e.g., it is necessary to know that 'm' refers to 'meters' when it follows a number; in this case, we would additionally include a TRICKY-WORDPLAY tag.

The annotation tagset of Poliak et al. (2018a) overlaps with ours in a few tags. For example, their 'pun' tag is a proper subset of our TRICKY-WORDPLAY tag. Their 'NER' and 'Gendered Anaphora' fall under our REFERENCE-COREFERENCE and REFERENCE-NAMES tags. Their recasting of the MegaVeridicality dataset (White and Rawlins, 2018) would have some overlap with our TRICKY-PRAGMATIC tag, for example, for the factive pair *Someone knew something happened.* ⇒ *something happened.*. Similarly, their examples recast from Schuler (2005, VerbNet) would likely recieve our TRICKY-SYNTACTIC tag for argument structure alternation, in at least some cases.

Rozen et al. (2019)'s tagset also has some overlap with ours, although none directly. They present two automatically generated datasets: one targets comparative reasoning about numbers—i.e., corresponding to a combination of our NUMERICAL-CARDINAL and BASIC-COMPARATIVESUPERLATIVE tags—and the other targets dative-alternation—which, like (Poliak et al., 2018a)'s recasting of VerbNet, would

Dataset		**Contexts**		**Statements**	
		Word$_{Len.}$	Sent.$_{Len.}$	Word$_{Len.}$	Sent.$_{Len.}$
ANLI	All	4.98 (0.60)	55.6 (13.7)	4.78 (0.76)	10.3 (5.28)
	A1	5.09 (**0.69**)	54.1 (8.35)	4.91 (0.74)	11.0 (5.36)
	A2	5.09 (0.47)	54.2 (8.24)	4.80 (0.77)	10.1 (4.95)
	A3	**4.73** (0.50)	**59.2** (**21.5**)	4.59 (0.76)	9.5 (5.38)
	C	5.00 (**0.79**)	55.8 (13.8)	4.76 (0.73)	**11.4 (6.51)**
	N	4.97 (0.47)	55.4 (13.8)	4.83 (0.78)	9.4 (4.49)
	E	5.00 (0.49)	55.7 (13.6)	4.75 (0.78)	10.3 (4.44)
MNLI	All	4.90 (0.97)	19.5 (13.6)	4.82 (0.90)	10.4 (4.43)
	M	**4.88 (1.10)**	19.3 (14.2)	4.78 (0.92)	9.9 (4.28)
	MM	**4.93 (0.87)**	19.7 (13.0)	4.86 (0.89)	10.8 (4.53)
	C	4.90 (0.97)	19.4 (13.6)	4.79 (0.90)	9.7 (3.99)
	N	4.90 (0.98)	19.4 (13.8)	4.79 (**0.85**)	10.9 (4.46)
	E	4.91 (0.96)	19.6 (13.5)	4.86 (**0.95**)	10.4 (4.71)
SNLI	All	4.31 (0.65)	14.0 (6.32)	4.23 (0.75)	7.5 (3.14)
	C	4.31 (0.64)	14.0 (6.35)	4.16 (0.71)	7.4 (2.90)
	N	4.31 (0.66)	13.8 (6.28)	4.26 (0.72)	8.3 (**3.36**)
	E	4.31 (0.64)	14.0 (6.31)	4.26 (**0.81**)	6.8 (2.90)

Table 11: Average length of words and sentences in contexts, statements, and reasons for ANLI, MultiNLI, SNLI. Average and (standard deviation).

Dataset	Word$_{Len.}$	Sent.$_{Len.}$	Count
All	4.54 (0.69)	21.05 (13.63)	3200
R1	**4.57** (0.65)	22.40 (13.80)	1000
R2	4.51 (**0.71**)	20.14 (12.96)	1000
R3	4.55 (0.70)	**20.81 (14.11)**	1200
C	4.53 (0.70)	19.46 (12.64)	1062
N	4.52 (0.64)	**23.81 (15.05)**	1066
E	**4.58 (0.72)**	19.87 (12.66)	1070
Numerical	4.44 (0.65)	21.79 (13.21)	1036
Basic	**4.63** (0.69)	21.31 (13.92)	1327
Reference	4.53 (0.70)	20.04 (13.01)	868
Tricky	4.56 (**0.71**)	20.58 (13.22)	893
Reasoning	4.52 (0.66)	**21.82 (14.08)**	1197
Imperfection	4.53 (**0.71**)	19.26 (13.06)	452

Table 12: Average length of words and sentences in rationales for ANLI. Average and (standard deviation).

probably correspond to our TRICKY-SYNTACTIC.

White et al. (2017) uses pre-existing semantic annotations to create an RTE/NLI formatted dataset. Their approach has several strong benefits, not the least of which is its use of minimal pairs to generate examples that can pinpoint exact failure points. For the first of our goals—understanding the contents of ANLI in particular—it would be interesting to have such annotations, and this could be a potentially fruitful future direction for research. But for the other—understanding current model performance on ANLI—it is not immediately clear to us that annotating ANLI for lexical semantic properties of predicates and their arguments (e.g., volition, awareness, and change of state) would help. In the end, it is an empirical question for future work.

From the above pairwise comparisons between existing annotation schemes (or data creation schemes), it should be clear there are many shared intuitions and many works are attempting to capture similar phenomena. We believe our tags thread the needle in a way that incorporates the best parts of the older annotation schemes while also innovating new phenomena and ways to view phenomena in relation to each other. In particular, very few of the schemes cited above arrange low level phenomena into a comprehensive multilevel hierarchy. This is one of the main benefits of our scheme. Our hierarchy allows us to compare models at multiple levels, and hopefully, as our models improve, it can allow us to explore transfer between different reasoning types.

B Dataset Properties

To further describe the ANLI dataset, we measure the length of words and sentences across all rounds and across all gold labels. We compare ANLI to SNLI and MNLI in Table 11. We also report length of rationales in Table 12. As the tables show, the statistics across classification labels are roughly the same within each dataset. It is easy to see that ANLI contains much longer contexts than both MNLI and SNLI. Overall, ANLI and MNLI appear more similar in statistics to each other than to SNLI, having have longer statements and longer words.

We analyzed the top 25 most frequent words (with stopwords removed based on the NLTK[17] stopword list) in development set contexts, statements, and rationales. We investigate frequent words for the entire dataset, by round, and by gold label (see Table 23), and by top-level annotation tag (see Table 24). The most frequent words in contexts reflect the domains of the original text. Since Wikipedia contexts were the most frequent in ANLI, words from Wikipedia including, for example 'film', 'album', 'directed', 'football', 'band', 'television' predictably figure prominently. References to nations, such as 'american', 'state', and 'national' are also common—perhaps reflecting a North American bias in the dataset.

Statements written by crowdworkers show a preference instead for terms like 'born', 'died', and 'people', suggesting again, that Wikipedia contexts, consisting largely of biographies, have a specific genre effect on constructed statements.

[17] https://www.nltk.org/

Model	A1	A2	A3	ANLI	hyperparameters
BERT (R1)	0	28	32	21	24-layer, 1024-hidden, 16-heads, 335M param.
RoBERTa (R2) Ens.	67	18	22	35	24-layer, 1024-hidden, 16-heads, 355M param.
RoBERTa (R3) Ens.	72	45	20	44	24-layer, 1024-hidden, 16-heads, 355M param.
RoBERTa-Large	74	51	46	56	24-layer, 1024-hidden, 16-heads, 355M param.
BART-Large	74	52	50	58	24-layer, 1024-hidden, 16-heads, 406M param.
XLNet-Large	74	52	51	58	24-layer, 1024-hidden, 16-heads, 340M param.
ELECTRA-Large	67	54	55	58	24-layer, 1024-hidden, 16-heads, 335M param.
ALBERT-XXLarge	**76**	**57**	**57**	**63**	12 repeating layer, 4096-hidden, 64-heads, 223M param.

Table 13: Accuracy for each model on the ANLI Development Sets (highest accuracy is bolded). Hyperparameters also provided. 'Ens.' refers to one model randomly selected from an ensemble of different seeds

Several examples appear in the top 25 most frequent words for both statements and contexts, including 'film', 'american', 'one', 'two', 'not', 'first', 'new', 'played', 'album', and 'city'. In particular, words such as 'one', 'first', 'new', and 'best' in contexts appear to be opposed by (near) antonyms such as 'two', 'last', 'old', 'least', and 'less' in statements. This suggests the words present in a context might affect how crowdworkers construct statements, potentially suggesting some lexical confounds in ANLI. Finally, we observe that the top 25 most frequent words in contexts are used roughly 3 times as often as the top 25 most frequent words in statements. This suggests that statements have wider and more varied vocabulary than contexts do.

B.1 Analyzing Annotator Rationales

We observe that the most frequent words in rationales differ from those in contexts and statements. The original annotators often use 'statement' and 'context' in their rationales to refer to example pairs, as well as 'system' to refer to the model; this last term is likely due to the fact that the name of the Mechanical Turk task used to employ crowdworkers in the original data collection was called "Beat the System" (Nie et al., 2020a, App. E). The set of most frequent words in rationales also contains, predictably, references to the model performance (e.g., 'correct', 'incorrect'), and to speech act verbs (e.g., 'says', 'states').

Interestingly, there is a higher number of verbs in the rationales denoting mental states (e.g., 'think', 'know', 'confused'), which suggests that the annotators could be ascribing theory of mind to the system, or at least using mental-state terms metaphorically—which could be due to the Nie et al. (2020a) data collection procedure that encourages crowdworkers to think of the model as an adversary. Rationales also contain more modals (e.g., 'probably', 'may', 'could'), which are often used to mark uncertainty, suggesting that the annotators are aware of the fact that their rationales might be biased by their human expectations. Finally, we note that the top 25 most frequent words used in rationales are much more common than are the top 25 most frequent words in contexts (by roughly two times) or in statements (by roughly 5-6 times). This suggests that vocabulary used for writing rationales is smaller than that in the contexts (from domains such as Wikipedia), and crowdworker annotated statements.

C Tag Breakdowns

Table 15 shows a breakdown of second-level tag incidence by top-level tag.

D Development Set Accuracies for 8 Transformer Models

Table 13 shows development set accuracies for all transformer models, by round. ANLI is still quite challenging, with even SOTA models barely exceeding 50% accuracy (although remember that the development set is approximately balanced 3-way classification, so we are beating random baseline). The ALBERT-XXLarge model achieves the highest accuracy on the full development set, reaching approximately 63% correct. On A1, the accuracy between the ALBERT-XXLarge and the other SOTA models hovers around two points, extending to 5–6 percentage points on A2, and 6–11 points on A3; the gap between ALBERT-XXLarge and the other SOTA models on the full ANLI development set hovers between 5 and 7 points.

E Model Predictions Breakdown by Tag

Model predictions by specific tags are in Table 16 (Basic), Table 17 (Numerical), Table 18 (Reasoning), Table 19 (Reference), Table 20 (Tricky), Table 21 (Imperfections).

For Numerical, Counting is the hardest, which makes sense given that Counting examples are relatively rare, and require that one actually counts phrases in the text, which is a metalinguistic skill. Ordinal is the next most difficult category, perhaps because, like Counting examples, Ordinal examples are relatively rare.[18] For

[18] Additionally, it seems difficult for models to bootstrap their Cardinal number knowledge for Ordinal numbers. One might hope that a model could bootstrap its knowledge of the order of cardinal numbers (e.g., that *one* comes before *two* and *three*) to perform well on their corresponding ordinals, However, numerical order information doesn't seem to be generally applied in these models. Perhaps this is because

BASIC, IMPLICATION, IDIOM and NEGATION were more difficult than LEXICAL, COMPARATIVE & SUPERLATIVE and COORDINATION. For REFERENCE, there is a lot of variation in the behavior of different models, particularly for the NAMES examples, although also for COREFERENCE examples, making it difficult to determine which is more difficult. Finally, for TRICKY examples, WORDPLAY examples the most difficult, again because these require complex metalinguistic abilities (i.e., word games, puns, and anagrams), but they are followed closely by EXHAUSTIFICATION examples, which require a complex type of pragmatic reasoning.[19]

F Model Predictions Breakdown by Domain

Table 22 shows the breakdown by genre. Wikipedia results correspond with the overall dataset: ALBERT-XXLarge performs the best on everything except TRICKY (where XLNET-LARGE performs best. ALBERT-XXLarge performs best nearly across the board on procedural text (being narrowly edged out by ELECTRA-Large on REASONING) and fiction (where ELECTRA-Large performs best on REFERENCE, and where BART-Large and ELECTRA-Large jointly take top slot for TRICKY). Finally, the news genre has the most variation: ALBERT-XXLarge still performs well on BASIC, TRICKY, REASONING tags, although ELECTRA-Large narrowly beats it on NUMERICAL; XLNet-Large beats out all others on REFERENCE in the news genre by 3+ points.

We aim to characterize relative performance between the models and note variation between model performances on different genres. For example, BART and RoBERTa struggle with fiction (except for on the TRICKY tag). For example, ELECTRA-Large performs quite well on NUMERICAL examples from the Wikipedia, news, and procedural datasets, but poorly on NUMERICAL examples from Fiction. Similary BART-LARGE performs well on TRICKY examples from Wikipedia, fiction, and news, but struggles with TRICKY examples in procedural text. To give a final example, RoBERTa-Large and XLNet-Large do well on REFERENCE examples in procedural text and Wikipediate to some extent (and, for XLNet-Large, also news text), but they struggle with fiction (and, for RoBERTa-Large, also news). Since models do not perform similarly on particular tags across genres, we suggest they have not learned fully generalizable knowledge corresponding to these tag types.[20]

many common ordinal numbers in English are not morphologically composed of their cardinal counterparts (e.g., *one* and *first*, *two* and *second*.

[19] See Chierchia et al. (2004) for a summary of the linguistic theory on exhaustification, although we adopt a wider definition of the phenomenon for the tag here as in Table 14.

[20] Although we analyze examples in the aggregate to abstract away from particular example idiosyncrasies, remember that examples can be tagged with any number of other inference types and may vary in many other features (e.g., length, vocabulary etc.), so they are not strictly comparable, and more work needs to be done to bolster these conclusions.

Top Level	Second Level	Third Level	Context	Hypothesis	Round	Label	Other Tags
Num.	Cardinal	Dates	Otryadyn Gündegmaa (…born **23 May** 1978), is a Mongolian sports shooter. …	Otryadyn Gündegmaa was born on May 23rd	A1	E (N)	Ordinal, Dates
		Ages	…John Fox probably won't roam an NFL sideline again…**the 63-year-old** Fox will now move into an analyst role…	John Fox **is under 60 years old**.	A3	C (E)	Ref., Coref.
	Ordinal	Dates	Black Robe…is a historical novel by Brian Moore set in New France in **the 17th century**…	Black Robe is a novel set in New France in **the mid 1600s**	A2	N (E)	Reasoning, Plaus., Likely, Cardinal
		Ages	John Barnard (**6 July 1794** at Chislehurst, Kent; **died 17 November 1878** at Cambridge, England) was an English amateur cricketer who played first-class cricket from 1815 to 1830. M…	John Barnard died before **his fifth birthday**.	A1	C (N)	Cardinal, Dates, Reasoning, Facts
	Counting		…The Demand Institute **was founded** in 2012 **by Mark Leiter and Jonathan Spector**…	The Demand Institute **was founded by two men**.	A2	E (N)	Ref., Names
	Nominal		Raúl Alberto Osella (born 8 June 1984 in Morteros) is an Argentine association footballer …He played **FIFA U-17 World Cup Final** for Argentina national team in **2001**. …	Raul Alberto Osella no longer plays for **the FIFA U-17 Argentina team**.	A2	E (N)	Reasoning, Facts, Tricky, Exhaust., Cardinal, Age, Dates
Basic	Lexical		…The **dating app Hater**, which matches users based on the things they hate, has compiled all of their data to create a map of **the foods everyone hates**…	Hater is an **app designed for foodies in relationships**.	A3	C (N)	
	Comp.& Super.		…try to hit your shot onto the upslope because they are **easier** putts to make **opposed to** downhill putts.	Upslope putts are **simple to do**	A3	N (E)	
	Implic.		[DANIDA]…provides humanitarian aid …to developing countries…	**Focusing on developing countries**, DANIDA hopes to improve citizens of different countries lives.	A2	E (N)	
	Idioms		…he set to work to hunt for his dear money…he found nothing; **all had been spent**…	The money **got up and walked away**.	A3	N (C)	Reasoning, Plaus., Unlikely
	Negation		Bernardo Provenzano …**was suspected of having been the head of the Corleonesi** …	It was **never confirmed that Bernardo Provenzano was the leader of the Corleonesi**.	A2	E (N)	Tricky, Prag.
	Coord.		…Dan went home and started cooking a steak. However, Dan accidentally **burned the steak**….	The steak was **cooked for too long or on too high a temperature**.	A3	E (N)	Basic, Lexical, Tricky, Prag.
Ref.	Coref.		…Tim was a tutor. …His latest student really pushed him, though. Tim could not get through to **him**. He had to give up…	Tim gave up on **her** eventually.	A3	C (E)	
	Names		**Never Shout Never** is an EP **by Never Shout Never** which was released December 8, 2009.…	**Never Shout Never** has a **self titled** EP.	A1	E (N)	
	Family		Sir Hugh Montgomery …was the **son** of Adam Montgomery, the 5th Laird of Braidstane, by his **wife and cousin**.	Sir Hugh Montgomery had at least one **sibling**.	A2	N (E)	Reasoning, Plaus., Likely
Tricky	Syntactic		Gunby…**is situated close to the borders with Leicestershire and Rutland**, and 9 mi south from **Grantham**…	Gunby **borders Rutland an Grantham**.	A1	C (E)	Imperfect., Spelling
	Prag.		…Singh won **the award** for Women Leadership in Industry…	…Singh won **many awards** for Women in Leadership in Industry.	A3	C (N)	
	Exhaust.		Linguistics …**involves an analysis of language form, language meaning, and language in context**. …	Form and meaning are **the only aspects** of language linguistics is concerned with.	A1	C (N)	
	Wordplay		…Brock Lesnar and Braun Strowman will both be under …on **Raw**…	**Raw is not an anagram of war**	A3	C (E)	
Reasoning	Plaus.	Likely	**B. Dalton Bookseller…founded in 1966 by Bruce Dayton**, a member of the same family that operated the Dayton's department store chain…	**Bruce Dayton founded the Dayton's department store chain**.	A1	C (E)	Ref., Names
		Unlikely	The Disenchanted Forest is a 1999 documentary film that follows endangered orphan orangutans …returned to their rainforest home. …	The Disenchanted Forest is …about **orangutans trying to learn how to fly by building their own planes**…	A2	C (N)	Reasoning, Facts
	Facts	Debatable	The Hitchhiker's Guide to the Galaxy is a 2005 British-American **comic science fiction film**…	Hitchhiker's Guide to the Galaxy is **a humorous film**.	A1	N (E)	Basic, Lexical
			…[Joey] decided to make [his mom] **pretend tea**. He got some hot water from the tap and mixed in the herb. But **to his shock**, his mom really drank the tea! **She said the herb he'd picked was chamomile**, a delicious tea!	**Joey knew how to make chamomile tea**.	A3	C (E)	
	Contain.	Parts	Milky Way Farm in Giles County, Tennessee, is **the former estate of Franklin C. Mars** …its manor house is now a venue for special events.	**The barn** is occassionaly staged for photo shoots.	A1	N (C)	Plaus., Unlikely, Imperfect., Spelling
		Loc.	Latin Jam Workout is a Latin Dance Fitness Program…[f]ounded in 2007 **in Los Angeles, California**, Latin Jam Workout combines …music with dance…	Latin Jam Workout was not created in **a latin american country**	A2	E (C)	Basic, Negation
		Times	Forbidden Heaven is a 1935 American drama film…released **on October 5, 1935** …	Forbidden Heaven is …film released **in the same month as the holiday Halloween**.	A1		Facts
Imperfect.	Error		Albert Levitt (March 14, 1887 – **June 18, 1968**) was a judge, law professor, attorney, and candidate for political office. …	Albert Levitt …held several positions in the legal field during his life, (**which ended in the summer of 1978**)…	A2	N (C)	Num., Cardinal, Dates
	Ambig.		Diablo is a 2015 Canadian-American psychological western …starring **Scott Eastwood**…It was the first Western starring **Eastwood**, the son of Western icon **Clint Eastwood**.	**It** was the last western starring **Eastwood**	A2	C (N)	Ref., Coref., Label, Basic, Comp.&Sup., Lexical, Num., Ordinal, Family
	Spelling		"Call My Name" is a song **recorded by Pietro Lombardi** from his first studio album "Jackpot"…It was **written and produced by** "DSDS" jury member Dieter Bohlen….	"Call my Name" was **written and recorded by Pierrot Lombardi** for his album "Jackpot".	A1	C (E)	Tricky, Syntactic, Imperfect., Spelling
	Translat.		Club **Deportivo Dénia** is a Spanish football team…it plays in **Divisiones Regionales de Fútbol** …holding home games at "**Estadio Diego Mena Cuesta**",…	Club **Deportivo Dénia** plays in the Spanish village "**Estadio Diego Mena Cuesta**".	A2	C (E)	Tricky, Syntactic

Table 14: Examples from the full scheme.

	Round	Overall	Cardinal	Ordinal	Counting	Nominal	Dates	Age
Numerical	A1	40.8%	37.8%	6.2%	1.9%	4.2%	27.4%	5.9%
	A2	38.5%	34.7%	6.7%	2.8%	3.5%	24.3%	6.7%
	A3	20.3%	18.6%	2.8%	2.3%	0.4%	7.1%	3.2%
	All	32.4%	29.6%	5.1%	2.3%	2.6%	18.8%	5.1%
	Round	Overall	Lexical	Compr. Supr.	Implic.	Idioms	Negation	Coord.
Basic	A1	31.4%	16.0%	5.3%	1.5%	0.3%	5.6%	5.5%
	A2	41.2%	20.2%	7.6%	2.4%	1.7%	9.8%	4.5%
	A3	50.2%	26.4%	4.9%	4.2%	2.2%	15.8%	6.1%
	All	41.5%	21.2%	5.9%	2.8%	1.4%	10.7%	5.4%
	Round	Overall	Coreference	Names	Family			
Ref. & Names	A1	24.5%	15.8%	12.5%	1.0%			
	A2	29.4%	22.7%	11.2%	1.7%			
	A3	27.5%	25.5%	1.9%	1.3%			
	All	27.1%	21.6%	8.1%	1.3%			
	Round	Overall	Syntactic	Prag.	Exhaustif.	Wordplay		
Tricky	A1	29.5%	14.5%	4.7%	5.5%	2.0%		
	A2	29.1%	8.0%	2.8%	8.6%	5.7%		
	A3	25.6%	9.3%	6.7%	4.8%	5.5%		
	All	27.9%	10.5%	4.8%	6.2%	4.5%		
	Round	Overall	Likely	Unlikely	Debatable	Facts	Containment	
Reasoning	A1	58.4%	25.7%	6.2%	3.1%	19.6%	11.0%	
	A2	62.7%	23.9%	6.9%	6.5%	25.6%	10.3%	
	A3	63.9%	22.7%	10.9%	10.8%	26.5%	5.3%	
	All	61.8%	24.0%	8.2%	7.0%	24.0%	8.7%	
	Round	Overall	Error	Ambiguous	EventCoref	Translation	Spelling	
Imperfections	A1	12.4%	3.3%	2.8%	0.9%	5.7%	5.8%	
	A2	13.5%	2.5%	4.0%	3.4%	6.2%	6.5%	
	A3	16.1%	2.2%	7.6%	1.9%	0.8%	5.5%	
	All	14.1%	2.6%	5.0%	2.1%	4.0%	5.9%	

Table 15: Percent examples in development set with particular tag, per round, on average.

BASIC **Round**	**Model**	**Basic**	Lexical	Comp.Sup.	ModusPonens	CauseEffect	Idiom	Negation	Coordination
A1	BERT (R1)	0.11 (0.56)	0.12 (0.59)	0.13 (0.66)	0.07 (0.31)	0.15 (0.55)	0.01 (0.45)	0.07 (0.40)	0.10 (0.52)
	RoBERTa Ensemble (R2)	0.69 (0.15)	0.73 (0.14)	0.63 (0.24)	0.43 (0.06)	**0.75** (0.02)	0.35 (0.12)	0.66 (0.17)	0.67 (0.13)
	RoBERTa Ensemble (R3)	0.72 (0.08)	0.78 (0.08)	0.72 (0.15)	0.32 (0.19)	**0.75** (0.01)	0.67 (0.02)	0.67 (0.06)	0.65 (0.08)
	RoBERTa-Large	0.76 (0.10)	**0.80** (0.12)	**0.82** (0.11)	**0.56** (0.08)	0.67 (0.20)	0.66 (0.02)	**0.71** (0.11)	0.74 (0.06)
	BART-Large	0.72 (**0.07**)	0.76 (**0.07**)	0.68 (0.08)	0.29 (**0.01**)	**0.75** (**0.00**)	**0.67** (**0.00**)	0.65 (**0.07**)	0.76 (0.11)
	XLNet-Large	0.75 (0.09)	0.78 (0.09)	0.77 (0.13)	0.23 (0.35)	**0.75** (**0.00**)	0.66 (0.02)	0.64 (0.11)	0.76 (**0.03**)
	ELECTRA-Large	0.68 (0.34)	0.71 (0.34)	0.71 (0.23)	0.39 (0.42)	0.60 (0.20)	0.31 (0.66)	0.61 (0.33)	0.65 (0.43)
	ALBERT-XXLarge	**0.76** (0.20)	**0.80** (0.19)	0.78 (0.24)	0.31 (0.46)	0.64 (0.15)	**0.67** (0.02)	0.63 (0.21)	**0.77** (0.14)
A2	BERT (R1)	0.29 (0.44)	0.31 (0.46)	0.31 (0.56)	0.24 (0.31)	0.29 (0.40)	0.35 (0.44)	0.24 (0.41)	0.20 (0.38)
	RoBERTa Ensemble (R2)	0.20 (0.25)	0.24 (0.23)	0.19 (0.33)	0.33 (0.32)	0.21 (0.35)	0.19 (0.21)	0.17 (0.26)	0.15 (0.29)
	RoBERTa Ensemble (R3)	0.41 (0.14)	0.43 (0.15)	0.49 (0.16)	0.55 (0.18)	0.15 (0.17)	0.28 (0.10)	0.42 (**0.09**)	0.41 (0.21)
	RoBERTa-Large	0.47 (0.17)	0.47 (0.17)	0.49 (0.23)	0.99 (0.07)	0.30 (0.23)	0.37 (0.10)	0.55 (0.12)	0.48 (0.15)
	BART-Large	0.48 (0.14)	0.55 (0.14)	0.48 (0.18)	0.40 (**0.00**)	0.23 (**0.06**)	0.43 (0.21)	0.48 (0.16)	0.44 (**0.09**)
	XLNet-Large	0.53 (**0.13**)	0.54 (**0.13**)	0.51 (**0.13**)	**0.80** (0.02)	0.39 (0.17)	0.53 (**0.02**)	0.56 (0.18)	**0.51** (**0.09**)
	ELECTRA-Large	0.52 (0.40)	0.54 (0.46)	0.46 (0.41)	0.47 (0.52)	0.38 (0.42)	0.53 (0.28)	0.56 (0.40)	0.56 (0.26)
	ALBERT-XXLarge	**0.58** (0.28)	**0.61** (0.28)	**0.53** (0.31)	**0.80** (0.04)	**0.48** (0.50)	**0.60** (0.31)	**0.64** (0.22)	0.50 (0.21)
A3	BERT (R1)	0.32 (0.50)	0.33 (0.51)	0.36 (0.59)	0.29 (0.72)	0.25 (0.57)	0.22 (0.47)	0.32 (0.46)	0.34 (0.50)
	RoBERTa Ensemble (R2)	0.26 (0.57)	0.26 (0.57)	0.29 (0.55)	0.25 (0.81)	0.16 (0.58)	0.24 (0.68)	0.25 (0.62)	0.26 (0.56)
	RoBERTa Ensemble (R3)	0.24 (0.53)	0.23 (0.53)	0.21 (0.53)	0.24 (0.57)	0.17 (0.51)	0.19 (0.57)	0.23 (0.57)	0.28 (0.50)
	RoBERTa-Large	0.45 (0.25)	0.44 (0.24)	0.46 (0.38)	0.45 (0.15)	0.39 (0.17)	0.42 (0.22)	0.46 (0.25)	0.49 (0.26)
	BART-Large	0.49 (**0.14**)	0.51 (0.16)	0.49 (**0.11**)	0.29 (**0.14**)	0.42 (**0.10**)	0.46 (**0.13**)	0.49 (**0.15**)	0.52 (0.13)
	XLNet-Large	0.49 (0.15)	0.50 (**0.12**)	0.47 (0.26)	0.34 (0.23)	0.40 (0.14)	0.44 (**0.13**)	0.46 (0.16)	**0.59** (**0.08**)
	ELECTRA-Large	0.52 (0.44)	0.56 (0.43)	0.51 (0.50)	0.58 (0.46)	0.43 (0.36)	**0.64** (0.48)	0.52 (0.43)	0.54 (0.44)
	ALBERT-XXLarge	**0.55** (0.36)	**0.55** (0.35)	**0.56** (0.48)	**0.65** (0.33)	**0.48** (0.27)	0.52 (0.44)	**0.56** (0.36)	0.53 (0.33)
ANLI	BERT (R1)	0.26 (0.50)	0.27 (0.51)	0.27 (0.60)	0.21 (0.50)	0.25 (0.52)	0.26 (0.46)	0.26 (0.44)	0.23 (0.48)
	RoBERTa Ensemble (R2)	0.34 (0.37)	0.36 (0.37)	0.35 (0.37)	0.33 (0.46)	0.25 (0.45)	0.23 (0.47)	0.29 (0.44)	0.36 (0.36)
	RoBERTa Ensemble (R3)	0.41 (0.30)	0.42 (0.31)	0.46 (0.27)	0.34 (0.36)	0.23 (0.35)	0.25 (0.36)	0.36 (0.35)	0.43 (0.29)
	RoBERTa-Large	0.53 (0.19)	0.54 (0.19)	0.57 (0.24)	0.61 (0.11)	0.40 (0.19)	0.42 (0.16)	0.53 (0.19)	0.57 (0.17)
	BART-Large	0.54 (**0.13**)	0.58 (0.13)	0.54 (**0.13**)	0.31 (**0.06**)	0.41 (0.08)	0.46 (0.15)	0.51 (**0.14**)	0.57 (0.11)
	XLNet-Large	0.56 (**0.13**)	0.58 (**0.11**)	0.57 (0.17)	0.41 (0.22)	0.44 (0.13)	0.49 (**0.08**)	0.52 (0.16)	**0.62** (**0.07**)
	ELECTRA-Large	0.56 (0.40)	0.59 (0.42)	0.54 (0.39)	0.49 (0.46)	0.44 (0.36)	**0.58** (0.42)	0.55 (0.40)	0.58 (0.39)
	ALBERT-XXLarge	**0.61** (0.30)	**0.63** (0.29)	**0.61** (0.34)	**0.58** (0.30)	**0.50** (0.32)	0.56 (0.37)	**0.60** (0.29)	0.60 (0.24)

Table 16: Correct label probability and entropy of label predictions for the BASIC subset: mean probability (mean entropy). BERT (R1) has zero accuracy, by construction, on A1 because it was used to collect A1, whereas RoBERTas (R2) and (R3) were part of an ensemble of several identical architectures with different random seeds, so they have low, but non-zero, accuracy on their respective rounds. Recall that the entropy for three equiprobable outcomes (i.e., random chance of three NLI labels) is upper bounded by ≈ 1.58.

NUMERICAL **Round**	**Model**	**Numerical**	Cardinal	Ordinal	Counting	Nominal	Dates	Age
A1	BERT (R1)	0.10 (0.57)	0.10 (0.57)	0.11 (0.60)	0.09 (0.64)	0.07 (0.46)	0.10 (0.58)	0.07 (0.41)
	RoBERTa Ensemble (R2)	0.68 (0.13)	0.68 (0.13)	0.71 (0.18)	0.51 (0.23)	0.72 (0.11)	0.69 (0.13)	0.64 (0.11)
	RoBERTa Ensemble (R3)	0.72 (**0.07**)	0.72 (**0.07**)	**0.77 (0.05)**	0.51 (0.23)	0.69 (0.06)	0.75 (**0.07**)	0.64 (**0.08**)
	RoBERTa-Large	0.73 (0.13)	0.73 (0.13)	0.75 (0.10)	0.58 (**0.10**)	0.76 (0.14)	0.74 (0.14)	0.65 (0.18)
	BART-Large	0.73 (0.10)	0.73 (0.10)	0.72 (0.11)	0.54 (0.12)	0.74 (**0.04**)	**0.77** (0.10)	0.67 (0.12)
	XLNet-Large	0.73 (0.10)	0.74 (0.10)	0.63 (0.08)	0.53 (0.15)	0.70 (0.11)	0.76 (0.09)	**0.71** (0.13)
	ELECTRA-Large	0.71 (0.29)	0.71 (0.28)	0.74 (0.35)	0.69 (0.42)	0.64 (0.23)	0.73 (0.27)	0.68 (0.38)
	ALBERT-XXLarge	**0.74** (0.22)	**0.75** (0.22)	0.72 (0.21)	0.56 (0.19)	**0.78** (0.19)	**0.77** (0.21)	**0.71** (0.32)
A2	BERT (R1)	0.29 (0.53)	0.28 (0.53)	0.33 (0.53)	0.43 (0.49)	0.31 (0.53)	0.25 (0.53)	0.18 (0.48)
	RoBERTa Ensemble (R2)	0.19 (0.28)	0.20 (0.28)	0.19 (0.24)	0.14 (0.30)	0.20 (0.34)	0.19 (0.26)	0.22 (0.25)
	RoBERTa Ensemble (R3)	0.50 (0.18)	0.51 (0.18)	0.50 (0.13)	0.36 (0.20)	0.44 (0.19)	0.55 (0.17)	0.51 (0.15)
	RoBERTa-Large	0.54 (0.22)	0.54 (0.22)	0.49 (0.21)	0.47 (0.17)	0.55 (**0.10**)	0.56 (0.24)	0.51 (0.26)
	BART-Large	0.55 (0.13)	0.54 (0.14)	0.57 (**0.10**)	0.56 (0.08)	0.47 (0.18)	0.56 (0.12)	0.50 (0.15)
	XLNet-Large	0.54 (**0.11**)	0.55 (**0.11**)	0.45 (0.14)	0.51 (0.06)	**0.54** (0.12)	0.57 (**0.11**)	0.54 (**0.14**)
	ELECTRA-Large	0.56 (0.36)	**0.57** (0.35)	0.55 (0.32)	0.52 (0.34)	0.49 (0.22)	**0.60** (0.36)	**0.59** (0.40)
	ALBERT-XXLarge	**0.57** (0.28)	**0.57** (0.28)	**0.60** (0.26)	**0.58** (0.26)	0.53 (0.20)	0.59 (0.30)	0.52 (0.34)
A3	BERT (R1)	0.34 (0.53)	0.34 (0.53)	0.43 (0.49)	0.34 (0.34)	0.41 (0.48)	0.31 (0.48)	0.28 (0.45)
	RoBERTa Ensemble (R2)	0.29 (0.47)	0.29 (0.46)	0.25 (0.47)	0.17 (0.48)	0.35 (0.41)	0.30 (0.34)	0.32 (0.36)
	RoBERTa Ensemble (R3)	0.20 (0.43)	0.20 (0.42)	0.25 (0.52)	0.11 (0.37)	0.20 (0.77)	0.22 (0.30)	0.26 (0.44)
	RoBERTa-Large	0.44 (0.32)	0.44 (0.32)	0.48 (0.29)	0.53 (0.15)	0.36 (0.53)	0.38 (0.33)	0.42 (0.37)
	BART-Large	0.51 (**0.14**)	0.52 (**0.14**)	0.48 (0.12)	0.51 (0.17)	0.59 (**0.07**)	0.51 (**0.10**)	0.46 (**0.14**)
	XLNet-Large	0.52 (0.15)	0.52 (0.16)	0.57 (**0.09**)	0.47 (**0.11**)	0.59 (**0.07**)	0.50 (0.17)	0.42 (0.16)
	ELECTRA-Large (tuned)	0.55 (0.46)	**0.56** (0.44)	0.52 (0.54)	0.58 (0.44)	**0.66** (0.53)	0.54 (0.43)	**0.57** (0.30)
	ALBERT-XXLarge	**0.56** (0.39)	**0.56** (0.38)	**0.61** (0.40)	**0.61** (0.32)	0.46 (0.43)	**0.58** (0.36)	0.56 (0.35)
A3	BERT (R1)	0.22 (0.54)	0.22 (0.55)	0.27 (0.54)	0.31 (0.48)	0.19 (0.49)	0.19 (0.54)	0.16 (0.45)
	RoBERTa Ensemble (R2)	0.41 (0.26)	0.41 (0.26)	0.40 (0.26)	0.25 (0.35)	0.48 (0.22)	0.44 (0.21)	0.39 (0.23)
	RoBERTa Ensemble (R3)	0.52 (0.20)	0.52 (0.19)	0.55 (0.18)	0.30 (0.27)	0.56 (0.16)	0.59 (0.14)	0.50 (0.19)
	RoBERTa-Large	0.59 (0.21)	0.59 (0.21)	0.59 (0.18)	0.52 (0.15)	0.65 (0.15)	0.62 (0.21)	0.54 (0.26)
	BART-Large	0.61 (**0.12**)	0.61 (**0.12**)	0.61 (0.11)	0.54 (0.12)	0.62 (**0.10**)	0.65 (**0.10**)	0.55 (**0.13**)
	XLNet-Large	0.61 (**0.12**)	0.62 (**0.12**)	0.54 (**0.10**)	0.50 (**0.10**)	0.62 (0.11)	0.65 (0.11)	0.57 (0.14)
	ELECTRA-Large	0.62 (0.35)	0.62 (0.34)	0.61 (0.38)	0.58 (0.40)	0.58 (0.25)	0.65 (0.33)	**0.62** (0.37)
	ALBERT-XXLarge	**0.64** (0.28)	**0.64** (0.28)	**0.65** (0.27)	**0.59** (0.26)	**0.66** (0.21)	**0.67** (0.27)	0.60 (0.33)

Table 17: Correct label probability and entropy of label predictions for the NUMERICAL subset: mean probability (mean entropy). BERT (R1) has zero accuracy, by construction, on A1 because it was used to collect A1, whereas RoBERTas (R2) and (R3) were part of an ensemble of several identical architectures with different random seeds, so they have low, but non-zero, accuracy on their respective rounds. Recall that the entropy for three equiprobable outcomes (i.e., random chance of three NLI labels) is upper bounded by ≈ 1.58.

REASONING							
Round	**Model**	**Reasoning**	Likely	Unlikely	Debatable	Facts	Containment
A1	BERT (R1)	0.13 (0.60)	0.14 (0.57)	0.15 (0.54)	0.16 (0.52)	0.11 (0.64)	0.11 (0.62)
	RoBERTa Ensemble (R2)	0.67 (0.13)	0.64 (0.16)	0.78 (0.13)	0.61 (0.05)	0.65 (0.12)	0.71 (0.14)
	RoBERTa Ensemble (R3)	0.73 (**0.08**)	0.72 (0.09)	0.78 (**0.04**)	0.68 (**0.00**)	0.71 (**0.08**)	0.75 (0.11)
	RoBERTa-Large	0.75 (0.12)	0.74 (0.15)	0.82 (0.09)	0.67 (0.06)	**0.74** (0.15)	0.75 (0.09)
	BART-Large	0.76 (**0.08**)	**0.78** (**0.07**)	0.86 (0.06)	**0.70** (0.04)	0.71 (0.09)	0.72 (0.12)
	XLNet-Large	0.74 (0.09)	0.74 (0.08)	0.82 (0.07)	**0.70** (0.09)	0.72 (0.12)	0.74 (**0.08**)
	ELECTRA-Large	0.66 (0.36)	0.68 (0.35)	0.77 (0.24)	0.66 (0.42)	0.63 (0.37)	0.58 (0.47)
	ALBERT-XXLarge	**0.77** (0.18)	0.77 (0.17)	**0.88** (0.09)	0.67 (0.21)	0.73 (0.21)	**0.76** (0.20)
A2	BERT (R1)	0.30 (0.47)	0.34 (0.44)	0.31 (0.42)	0.36 (0.44)	0.23 (0.49)	0.33 (0.54)
	RoBERTa Ensemble (R2)	0.21 (0.26)	0.27 (0.28)	0.21 (0.33)	0.16 (0.27)	0.18 (0.22)	0.17 (0.19)
	RoBERTa Ensemble (R3)	0.43 (0.16)	0.43 (0.14)	0.45 (0.18)	0.43 (0.16)	0.40 (0.13)	0.38 (0.17)
	RoBERTa-Large	0.51 (0.21)	0.48 (0.19)	0.56 (0.20)	0.43 (0.21)	0.49 (0.23)	0.49 (0.22)
	BART-Large	0.52 (0.13)	0.61 (**0.12**)	0.53 (0.13)	0.48 (0.13)	0.43 (0.14)	0.48 (0.17)
	XLNet-Large	0.53 (**0.12**)	0.57 (0.13)	0.56 (**0.12**)	0.49 (**0.05**)	0.48 (**0.11**)	0.49 (**0.11**)
	ELECTRA-Large	0.53 (0.40)	0.58 (0.39)	0.54 (0.38)	0.52 (0.39)	0.49 (0.39)	0.51 (0.42)
	ALBERT-XXLarge	**0.57** (0.29)	**0.62** (0.27)	**0.65** (0.30)	**0.55** (0.25)	**0.50** (0.30)	**0.53** (0.29)
A3	BERT (R1)	0.34 (0.51)	0.37 (0.47)	0.38 (0.48)	0.35 (0.51)	0.29 (0.54)	0.35 (0.46)
	RoBERTa Ensemble (R2)	0.26 (0.54)	0.25 (0.51)	0.28 (0.58)	0.25 (0.62)	0.25 (0.51)	0.28 (0.38)
	RoBERTa Ensemble (R3)	0.23 (0.50)	0.23 (0.47)	0.25 (0.52)	0.21 (0.56)	0.22 (0.48)	0.20 (0.38)
	RoBERTa-Large	0.44 (0.26)	0.44 (0.25)	0.51 (0.25)	0.47 (0.24)	0.40 (0.27)	0.50 (0.32)
	BART-Large	0.50 (**0.14**)	0.52 (0.14)	0.57 (**0.13**)	0.47 (**0.15**)	0.44 (**0.14**)	0.58 (0.16)
	XLNet-Large	0.49 (**0.14**)	0.47 (**0.13**)	0.56 (0.14)	0.50 (0.16)	0.47 (0.15)	0.51 (**0.13**)
	ELECTRA-Large	0.51 (0.45)	0.49 (0.48)	0.56 (0.39)	0.49 (0.49)	0.48 (0.44)	0.51 (0.48)
	ALBERT-XXLarge	**0.57** (0.33)	**0.59** (0.33)	**0.65** (0.32)	**0.58** (0.37)	**0.50** (0.33)	**0.55** (0.23)
ANLI	BERT (R1)	0.26 (0.52)	0.29 (0.49)	0.31 (0.48)	0.33 (0.49)	0.23 (0.55)	0.25 (0.56)
	RoBERTa Ensemble (R2)	0.37 (0.33)	0.39 (0.32)	0.38 (0.41)	0.28 (0.44)	0.33 (0.32)	0.41 (0.21)
	RoBERTa Ensemble (R3)	0.44 (0.27)	0.46 (0.24)	0.43 (0.32)	0.34 (0.37)	0.41 (0.26)	0.48 (0.19)
	RoBERTa-Large	0.55 (0.20)	0.55 (0.19)	0.60 (0.20)	0.49 (0.21)	0.52 (0.22)	0.60 (0.19)
	BART-Large	0.58 (**0.12**)	0.63 (**0.11**)	0.63 (**0.12**)	0.51 (0.13)	0.50 (**0.13**)	0.60 (0.15)
	XLNet-Large	0.58 (**0.12**)	0.59 (0.12)	0.62 (**0.12**)	0.52 (0.12)	0.54 (**0.13**)	0.59 (**0.10**)
	ELECTRA-Large	0.56 (0.40)	0.58 (0.41)	0.60 (0.35)	0.52 (0.45)	0.52 (0.41)	0.54 (0.46)
	ALBERT-XXLarge	**0.63** (0.27)	**0.66** (0.26)	**0.70** (0.26)	**0.58** (0.31)	**0.56** (0.29)	**0.63** (0.24)

Table 18: Correct label probability and entropy of label predictions for the REASONING subset: mean probability (mean entropy). BERT (R1) has zero accuracy, by construction, on A1 because it was used to collect A1, whereas RoBERTas (R2) and (R3) were part of an ensemble of several identical architectures with different random seeds, so they have low, but non-zero, accuracy on their respective rounds. Recall that the entropy for three equiprobable outcomes (i.e., random chance of three NLI labels) is upper bounded by ≈ 1.58.

REFERENCE					
Round	**Model**	**Reference**	Coreference	Names	Family
A1	BERT (R1)	0.12 (0.59)	0.11 (0.56)	0.12 (0.60)	0.12 (0.56)
	RoBERTa Ensemble (R2)	0.66 (0.15)	0.67 (0.15)	0.68 (0.15)	0.29 (0.19)
	RoBERTa Ensemble (R3)	0.70 (0.08)	0.70 (**0.08**)	0.75 (0.06)	0.44 (0.17)
	RoBERTa-Large	0.75 (0.15)	0.76 (0.15)	0.77 (0.15)	0.52 (0.26)
	BART-Large	0.70 (0.11)	0.73 (0.13)	0.73 (**0.09**)	0.54 (0.10)
	XLNet-Large	0.72 (**0.09**)	0.74 (**0.08**)	0.75 (**0.09**)	0.62 (**0.09**)
	ELECTRA-Large	0.63 (0.41)	0.64 (0.41)	0.66 (0.40)	0.61 (0.35)
	ALBERT-XXLarge	**0.77** (0.18)	**0.78** (0.18)	**0.80** (0.17)	**0.67** (0.12)
A2	BERT (R1)	0.31 (0.47)	0.29 (0.47)	0.33 (0.48)	0.34 (0.41)
	RoBERTa Ensemble (R2)	0.19 (0.24)	0.20 (0.24)	0.16 (0.24)	0.18 (0.24)
	RoBERTa Ensemble (R3)	0.45 (0.14)	0.46 (0.16)	0.42 (0.14)	0.45 (0.17)
	RoBERTa-Large	0.49 (0.20)	0.53 (0.20)	0.42 (0.19)	0.44 (0.16)
	BART-Large	0.50 (0.13)	0.52 (0.13)	0.41 (0.13)	0.40 (**0.14**)
	XLNet-Large	0.50 (**0.10**)	0.52 (**0.10**)	0.43 (**0.08**)	0.48 (0.19)
	ELECTRA-Large	0.53 (0.38)	0.55 (0.39)	0.48 (0.39)	0.38 (0.47)
	ALBERT-XXLarge	**0.56** (0.25)	**0.58** (0.28)	**0.49** (0.22)	**0.58** (0.17)
A3	BERT (R1)	0.32 (0.49)	0.33 (0.48)	0.27 (0.51)	0.25 (0.59)
	RoBERTa Ensemble (R2)	0.27 (0.55)	0.27 (0.53)	0.26 (0.76)	0.39 (0.39)
	RoBERTa Ensemble (R3)	0.25 (0.54)	0.24 (0.54)	0.26 (0.46)	0.47 (0.41)
	RoBERTa-Large	0.46 (0.27)	0.46 (0.27)	0.46 (0.38)	0.47 (0.22)
	BART-Large	0.50 (**0.14**)	0.49 (**0.13**)	0.67 (0.17)	**0.62** (0.23)
	XLNet-Large	0.52 (0.15)	0.50 (0.16)	**0.70** (**0.15**)	0.61 (**0.09**)
	ELECTRA-Large	0.52 (0.48)	0.51 (0.48)	0.66 (0.42)	0.51 (0.45)
	ALBERT-XXLarge	**0.54** (0.32)	**0.53** (0.32)	0.66 (0.31)	**0.62** (0.27)
ANLI	BERT (R1)	0.26 (0.51)	0.27 (0.49)	0.22 (0.54)	0.25 (0.51)
	RoBERTa Ensemble (R2)	0.35 (0.33)	0.34 (0.35)	0.42 (0.24)	0.29 (0.28)
	RoBERTa Ensemble (R3)	0.45 (0.28)	0.42 (0.31)	0.56 (0.13)	0.46 (0.26)
	RoBERTa-Large	0.56 (0.21)	0.55 (0.22)	0.59 (0.19)	0.47 (0.20)
	BART-Large	0.55 (0.13)	0.55 (0.13)	0.59 (**0.12**)	0.51 (0.16)
	XLNet-Large	0.57 (**0.12**)	0.56 (**0.12**)	0.61 (0.09)	0.56 (**0.13**)
	ELECTRA-Large	0.55 (0.43)	0.55 (0.43)	0.59 (0.40)	0.48 (0.43)
	ALBERT-XXLarge	**0.61** (0.25)	**0.60** (0.27)	**0.65** (0.20)	**0.62** (0.20)

Table 19: Correct label probability and entropy of label predictions for the REFERENCE subset: mean probability (mean entropy). BERT (R1) has zero accuracy, by construction, on A1 because it was used to collect A1, whereas RoBERTas (R2) and (R3) were part of an ensemble of several identical architectures with different random seeds, so they have low, but non-zero, accuracy on their respective rounds. Recall that the entropy for three equiprobable outcomes (i.e., random chance of three NLI labels) is upper bounded by ≈ 1.58.

TRICKY Round	**Model**	**Tricky**	Syntactic	Pragmatic	Exhaustification	Wordplay
A1	BERT (R1)	0.10 (0.56)	0.10 (0.54)	0.09 (0.56)	0.11 (0.56)	0.13 (0.72)
	RoBERTa Ensemble (R2)	0.60 (0.18)	0.60 (0.17)	0.60 (0.23)	0.59 (0.17)	0.52 (0.15)
	RoBERTa Ensemble (R3)	0.65 (0.09)	0.67 (**0.09**)	**0.72** (0.08)	0.54 (0.11)	0.51 (0.06)
	RoBERTa-Large	**0.70** (0.14)	0.72 (0.15)	0.68 (0.10)	**0.64** (0.13)	0.65 (0.15)
	BART-Large	**0.70 (0.08)**	**0.73 (0.09)**	0.64 (0.07)	0.62 (**0.08**)	0.75 (**0.02**)
	XLNet-Large	**0.70** (0.10)	**0.73** (0.11)	0.66 (**0.06**)	0.56 (0.10)	**0.78** (0.15)
	ELECTRA-Large	0.62 (0.44)	0.62 (0.49)	0.62 (0.40)	0.56 (0.41)	0.60 (0.45)
	ALBERT-XXLarge	0.65 (0.21)	0.66 (0.19)	0.61 (0.17)	0.58 (0.25)	0.63 (0.23)
A2	BERT (R1)	0.25 (0.48)	0.22 (0.53)	0.20 (0.35)	0.29 (0.47)	0.21 (0.47)
	RoBERTa Ensemble (R2)	0.16 (0.23)	0.19 (0.25)	0.10 (0.13)	0.20 (0.21)	0.09 (0.30)
	RoBERTa Ensemble (R3)	0.44 (0.14)	0.40 (**0.13**)	0.33 (**0.10**)	0.37 (0.16)	0.59 (0.14)
	RoBERTa-Large	0.48 (0.22)	**0.49** (0.20)	0.33 (0.21)	0.40 (0.25)	0.59 (0.16)
	BART-Large	0.48 (0.15)	0.46 (0.14)	0.26 (0.14)	0.45 (0.15)	0.58 (0.13)
	XLNet-Large	**0.52 (0.12)**	0.48 (**0.13**)	0.39 (0.14)	0.50 (**0.14**)	**0.60 (0.07)**
	ELECTRA-Large	0.51 (0.45)	**0.49** (0.52)	0.39 (0.44)	0.47 (0.41)	0.57 (0.45)
	ALBERT-XXLarge	0.50 (0.26)	0.44 (0.25)	**0.40** (0.28)	**0.51** (0.29)	0.42 (0.24)
A3	BERT (R1)	0.29 (0.55)	0.29 (0.50)	0.29 (0.64)	0.28 (0.48)	0.25 (0.58)
	RoBERTa Ensemble (R2)	0.24 (0.58)	0.26 (0.51)	0.24 (0.62)	0.18 (0.53)	0.24 (0.72)
	RoBERTa Ensemble (R3)	0.25 (0.54)	0.29 (0.53)	0.20 (0.57)	0.23 (0.58)	0.24 (0.50)
	RoBERTa-Large	0.49 (0.25)	0.47 (0.28)	0.41 (0.19)	0.46 (0.24)	0.63 (0.25)
	BART-Large	0.53 (0.18)	0.51 (0.17)	0.43 (0.21)	0.46 (0.18)	**0.72** (0.20)
	XLNet-Large	0.51 (**0.14**)	**0.57 (0.14)**	**0.46 (0.13)**	0.36 (**0.11**)	0.57 (**0.16**)
	ELECTRA-Large	**0.54** (0.44)	0.53 (0.44)	0.41 (0.50)	**0.49** (0.45)	**0.72** (0.41)
	ALBERT-XXLarge	0.52 (0.32)	0.53 (0.36)	**0.46** (0.30)	0.44 (0.31)	0.62 (0.32)
ANLI	BERT (R1)	0.21 (0.53)	0.19 (0.52)	0.22 (0.56)	0.24 (0.50)	0.22 (0.55)
	RoBERTa Ensemble (R2)	0.33 (0.34)	0.39 (0.30)	0.32 (0.41)	0.30 (0.29)	0.22 (0.47)
	RoBERTa Ensemble (R3)	0.45 (0.26)	0.48 (0.24)	0.38 (0.34)	0.38 (0.27)	0.42 (0.29)
	RoBERTa-Large	0.56 (0.20)	0.58 (0.21)	0.48 (0.17)	0.48 (0.21)	0.62 (0.20)
	BART-Large	**0.57** (0.14)	0.59 (0.13)	0.46 (0.15)	0.50 (0.14)	**0.67** (0.15)
	XLNet-Large	**0.57 (0.12)**	**0.62 (0.12)**	**0.51 (0.11)**	0.48 (**0.12**)	0.61 (**0.12**)
	ELECTRA-Large	0.56 (0.44)	0.56 (0.48)	0.47 (0.46)	0.50 (0.42)	0.65 (0.43)
	ALBERT-XXLarge	0.56 (0.26)	0.57 (0.26)	0.49 (0.26)	**0.51** (0.28)	0.54 (0.28)

Table 20: Correct label probability and entropy of label predictions for the TRICKY subset: mean probability (mean entropy). BERT (R1) has zero accuracy, by construction, on A1 because it was used to collect A1, whereas RoBERTas (R2) and (R3) were part of an ensemble of several identical architectures with different random seeds, so they have low, but non-zero, accuracy on their respective rounds. Recall that the entropy for three equiprobable outcomes (i.e., random chance of three NLI labels) is upper bounded by ≈ 1.58.

IMPERFECTIONS Round	Model	Imperfections	Errors	Ambiguity	EventCoref	Translation	Spelling
A1	BERT (R1)	0.13 (0.57)	0.07 (0.38)	0.17 (0.73)	0.12 (0.77)	0.11 (0.59)	0.14 (0.64)
	RoBERTa Ensemble (R2)	0.61 (0.14)	0.38 (0.11)	0.53 (0.19)	0.82 (0.25)	0.67 (0.17)	0.77 (0.12)
	RoBERTa Ensemble (R3)	0.68 (**0.07**)	0.49 (0.12)	0.57 (**0.02**)	**0.89 (0.00)**	0.71 (0.06)	0.81 (0.07)
	RoBERTa-Large	0.68 (0.13)	0.46 (0.15)	0.65 (0.17)	0.88 (0.04)	**0.75** (0.13)	0.79 (0.14)
	BART-Large	0.71 (0.08)	**0.52 (0.06)**	**0.73** (0.10)	0.78 (**0.00**)	0.74 (**0.11**)	0.79 (0.11)
	XLNet-Large	0.67 (0.08)	0.49 (0.11)	0.58 (0.18)	0.81 (0.24)	0.72 (0.06)	**0.81 (0.06)**
	ELECTRA-Large	0.63 (0.40)	0.49 (0.43)	0.65 (0.51)	0.58 (0.50)	0.73 (0.37)	0.70 (0.35)
	ALBERT-XXLarge	**0.69** (0.22)	0.48 (0.24)	0.62 (0.27)	0.78 (0.19)	0.71 (0.19)	0.78 (0.21)
A2	BERT (R1)	0.33 (0.48)	0.42 (0.39)	0.32 (0.47)	0.27 (0.43)	0.29 (0.51)	0.34 (0.45)
	RoBERTa Ensemble (R2)	0.19 (0.27)	0.22 (0.22)	0.19 (0.23)	0.21 (0.33)	0.16 (0.23)	0.21 (0.28)
	RoBERTa Ensemble (R3)	0.33 (0.14)	0.34 (0.17)	0.43 (0.11)	0.40 (**0.11**)	0.46 (0.13)	0.32 (0.12)
	RoBERTa-Large	0.49 (0.19)	0.38 (0.26)	0.50 (0.16)	0.50 (0.20)	0.48 (0.27)	0.56 (0.18)
	BART-Large	0.42 (**0.10**)	0.29 (0.08)	0.48 (**0.10**)	**0.58** (0.12)	0.48 (0.13)	0.45 (0.11)
	XLNet-Large	0.44 (**0.10**)	0.36 (**0.03**)	0.48 (**0.10**)	0.54 (0.13)	0.55 (**0.10**)	0.45 (**0.10**)
	ELECTRA-Large	0.54 (0.39)	0.40 (0.24)	0.63 (0.33)	0.55 (0.38)	0.56 (0.44)	**0.60** (0.46)
	ALBERT-XXLarge	**0.58** (0.32)	**0.60** (0.42)	**0.69** (0.26)	0.54 (0.23)	**0.66** (0.26)	0.54 (0.30)
A3	BERT (R1)	0.31 (0.54)	0.30 (0.57)	0.28 (0.58)	0.24 (0.29)	0.42 (0.76)	0.36 (0.52)
	RoBERTa Ensemble (R2)	0.23 (0.58)	0.22 (0.65)	0.23 (0.58)	0.36 (0.52)	0.26 (0.21)	0.19 (0.46)
	RoBERTa Ensemble (R3)	0.23 (0.52)	0.23 (0.55)	0.17 (0.52)	0.32 (0.48)	0.16 (0.26)	0.22 (0.46)
	RoBERTa-Large	0.40 (0.23)	0.32 (0.14)	0.35 (0.19)	0.70 (0.18)	**0.56** (0.13)	0.39 (0.24)
	BART-Large	0.48 (0.17)	0.37 (0.13)	0.39 (0.17)	0.63 (0.26)	0.30 (**0.03**)	**0.53** (0.15)
	XLNet-Large	0.43 (**0.14**)	**0.41 (0.10)**	0.40 (**0.15**)	0.64 (**0.13**)	0.52 (0.19)	0.40 (**0.12**)
	ELECTRA-Large	0.47 (0.49)	0.32 (0.42)	0.43 (0.53)	0.63 (0.37)	0.33 (0.40)	0.48 (0.46)
	ALBERT-XXLarge	**0.52** (0.33)	0.39 (0.31)	**0.50** (0.36)	**0.68** (0.32)	0.47 (0.49)	0.49 (0.28)
ANLI	BERT (R1)	0.27 (0.53)	0.24 (0.44)	0.27 (0.58)	0.24 (0.43)	0.22 (0.57)	0.28 (0.53)
	RoBERTa Ensemble (R2)	0.32 (0.37)	0.28 (0.31)	0.27 (0.42)	0.35 (0.39)	0.39 (0.20)	0.38 (0.29)
	RoBERTa Ensemble (R3)	0.39 (0.28)	0.36 (0.27)	0.31 (0.33)	0.44 (0.22)	0.55 (0.11)	0.44 (0.22)
	RoBERTa-Large	0.50 (0.19)	0.39 (0.18)	0.44 (0.18)	0.62 (0.17)	0.60 (0.20)	0.57 (0.19)
	BART-Large	0.52 (0.12)	0.41 (**0.09**)	0.47 (0.14)	0.62 (0.15)	0.58 (0.11)	0.58 (0.12)
	XLNet-Large	0.50 (**0.11**)	0.42 (**0.09**)	0.45 (**0.14**)	0.61 (**0.14**)	0.62 (**0.09**)	0.55 (**0.10**)
	ELECTRA-Large	0.54 (0.44)	0.41 (0.37)	0.52 (0.48)	0.58 (0.40)	0.62 (0.41)	0.59 (0.42)
	ALBERT-XXLarge	**0.59** (0.30)	**0.49** (0.32)	**0.57** (0.32)	**0.62** (0.26)	**0.67** (0.25)	**0.60** (0.27)

Table 21: Correct label probability and entropy of label predictions for the IMPERFECTIONS subset: mean probability (mean entropy). BERT (R1) has zero accuracy, by construction, on A1 because it was used to collect A1, whereas RoBERTas (R2) and (R3) were part of an ensemble of several identical architectures with different random seeds, so they have low, but non-zero, accuracy on their respective rounds. Recall that the entropy for three equiprobable outcomes (i.e., random chance of three NLI labels) is upper bounded by ≈ 1.58. A3 had no examples of TRANSLATION, so no numbers can be reported.

Genre	Model	Numerical	Basic	Reference	Tricky	Reasoning	Imperfections
Wikipedia	BERT (R1)	0.20 (0.55)	0.23 (0.49)	0.24 (0.51)	0.18 (0.52)	0.23 (0.53)	0.24 (0.52)
	RoBERTa Ensemble (R2)	0.43 (0.21)	0.40 (0.21)	0.40 (0.21)	0.37 (0.22)	0.42 (0.21)	0.37 (0.21)
	RoBERTa Ensemble (R3)	0.58 (0.13)	0.51 (0.12)	0.54 (0.12)	0.52 (0.12)	0.53 (0.13)	0.46 (0.12)
	RoBERTa-Large	0.61 (0.18)	0.57 (0.15)	0.59 (0.18)	0.58 (0.18)	0.60 (0.18)	0.55 (0.18)
	BART-Large	0.63 (**0.11**)	0.57 (**0.11**)	0.58 (0.12)	0.58 (0.12)	0.62 (**0.11**)	0.54 (0.10)
	XLNet-Large	0.62 (**0.11**)	0.61 (0.12)	0.59 (**0.09**)	**0.60 (0.11)**	0.62 (**0.11**)	0.53 (**0.09**)
	ELECTRA-Large	0.62 (0.33)	0.57 (0.38)	0.57 (0.40)	0.56 (0.44)	0.58 (0.38)	0.57 (0.40)
	ALBERT-XXLarge	**0.65** (0.25)	**0.64** (0.25)	**0.65** (0.22)	0.56 (0.24)	**0.66** (0.24)	**0.63** (0.27)
Fiction	BERT (R1)	0.49 (0.35)	0.28 (0.54)	0.29 (0.52)	0.35 (0.60)	0.29 (0.51)	0.30 (0.62)
	RoBERTa Ensemble (R2)	0.32 (0.73)	0.25 (0.68)	0.26 (0.70)	0.24 (0.71)	0.26 (0.63)	0.24 (0.73)
	RoBERTa Ensemble (R3)	0.35 (0.55)	0.26 (0.70)	0.29 (0.73)	0.26 (0.72)	0.27 (0.64)	0.28 (0.73)
	RoBERTa-Large	0.41 (0.14)	0.46 (0.22)	0.45 (0.26)	0.56 (0.16)	0.45 (0.24)	0.35 (0.15)
	BART-Large	0.14 (0.06)	0.49 (0.17)	0.46 (0.14)	**0.59** (0.12)	0.48 (0.14)	0.47 (0.14)
	XLNet-Large	0.57 (**0.01**)	0.49 (**0.08**)	0.50 (**0.10**)	0.52 (**0.09**)	0.52 (**0.10**)	0.40 (**0.04**)
	ELECTRA-Large	0.23 (0.28)	0.54 (0.36)	**0.56** (0.45)	**0.59** (0.36)	0.51 (0.38)	0.47 (0.45)
	ALBERT-XXLarge	**0.65** (0.27)	**0.55** (0.27)	0.50 (0.23)	0.52 (0.26)	**0.61** (0.28)	**0.62** (0.34)
News	BERT (R1)	0.38 (0.47)	0.32 (0.53)	0.26 (0.48)	0.25 (0.61)	0.40 (0.49)	0.39 (0.46)
	RoBERTa Ensemble (R2)	0.23 (0.40)	0.24 (0.43)	0.16 (0.32)	0.23 (0.49)	0.26 (0.41)	0.14 (0.64)
	RoBERTa Ensemble (R3)	0.19 (0.30)	0.22 (0.37)	0.21 (0.34)	0.26 (0.40)	0.22 (0.39)	0.23 (0.41)
	RoBERTa-Large	0.43 (0.31)	0.46 (0.22)	0.41 (**0.14**)	0.49 (0.15)	0.47 (0.23)	0.50 (0.23)
	BART-Large	0.56 (0.16)	0.49 (0.14)	0.41 (0.18)	0.63 (0.17)	0.54 (0.15)	**0.66** (0.20)
	XLNet-Large	0.56 (**0.14**)	0.51 (**0.13**)	**0.55** (0.18)	0.52 (**0.12**)	0.49 (**0.14**)	0.48 (**0.17**)
	ELECTRA-Large	**0.68** (0.39)	0.53 (0.39)	0.45 (0.33)	0.57 (0.35)	0.48 (0.40)	0.53 (0.45)
	ALBERT-XXLarge	0.67 (0.32)	**0.56** (0.22)	0.52 (0.23)	**0.64** (0.19)	**0.55** (0.24)	0.60 (0.26)
Procedural	BERT (R1)	0.37 (0.43)	0.30 (0.57)	0.38 (0.48)	0.19 (0.46)	0.34 (0.56)	0.30 (0.58)
	RoBERTa Ensemble (R2)	0.28 (0.65)	0.24 (0.67)	0.22 (0.69)	0.21 (0.70)	0.26 (0.70)	0.23 (0.60)
	RoBERTa Ensemble (R3)	0.21 (0.63)	0.24 (0.59)	0.21 (0.68)	0.27 (0.64)	0.25 (0.63)	0.25 (0.51)
	RoBERTa-Large	0.58 (0.23)	0.50 (0.13)	0.65 (0.25)	**0.57** (0.25)	0.45 (0.20)	0.45 (**0.07**)
	BART-Large	0.53 (**0.08**)	0.47 (**0.07**)	0.49 (**0.19**)	0.41 (**0.16**)	0.47 (**0.10**)	0.52 (0.09)
	XLNet-Large	0.57 (0.10)	0.53 (0.14)	0.66 (0.21)	0.53 (0.17)	0.49 (0.15)	**0.57** (0.18)
	ELECTRA-Large	**0.67** (0.35)	0.58 (0.43)	0.58 (0.44)	0.55 (0.41)	**0.58** (0.44)	0.42 (0.52)
	ALBERT-XXLarge	0.66 (0.26)	**0.61** (0.32)	**0.71** (0.29)	**0.57** (0.29)	0.56 (0.31)	0.53 (0.26)

Table 22: Probability of the correct label (entropy of label predictions) for each model on each top level annotation tag. BERT (R1) has zero accuracy, by construction, on A1 because it was used to collect A1, whereas RoBERTas (R2) and (R3) were part of an ensemble of several identical architectures with different random seeds, so they have low, but non-zero, accuracy on their respective rounds. Recall that the entropy for three equiprobable outcomes (i.e., random chance of three NLI labels) is upper bounded by ≈ 1.58.

Subset	Context	Statement	Rationale	Context+Statement
ANLI	film (647), american (588), **known** (377), first (376), (born (365), **also** (355), one (342), new (341), released (296), album (275), **united** (249), **directed** (240), not (236), – (218), **based** (214), **series** (196), **best** (191), may (188), **band** (185), **state** (182), **football** (177), two (175), written (175), **television** (175), **national** (169), **south** (165)	not (252), born (132), years (120), released (107), one (87), film (83), first (82), only (76), **people** (75), year (61), **played** (58), new (58), two (54), **made** (54), album (49), no (46), **died** (46), **won** (46), **less** (44), **last** (42), american (41), **years.** (40), **three** (40), written (38), **used** (37), **john** (37)	not (1306), **system** (753), **statement** (494), **know** (343), **think** (274), **definitely** (268), **context** (261), **correct** (243), **difficult** (228), only (224), **doesn't** (223), may (221), **confused** (218), no (200), **says** (198), **incorrect** (193), **text** (184), **could** (181), **states** (166), born (160), one (155), **say** (147), years (146), **don't** (140), **would** (130), **whether** (129)	film (730), american (629), not (488), first (458), one (429), known (414), released (403), new (399), also (379), (born (368), album (324), united (281), directed (274), based (238), two (229), born (226), series (223), played (221), – (221), best (220), band (219), only (213), written (213), football (208), may (208), state (204)
R1	film (299), american (272), known (175), (born (169), first (158), also (129), released (119), album (115), directed (106), based (104), united (103), new (97), – (93), football (88), one (84), band (77), best (77), south (73), **former** (71), written (70), series (67), played (67), **march** (66), city (65), located (65), television (64)	born (65), film (47), not (46), years (45), released (43), first (36), died (26), only (25), american (24), **population** (23), **old** (23), album (22), won (22), played (21), directed (21), new (19), last (18), football (18), **century.** (18), year (18), united (17), years. (16), **world** (16), written (16), one (16), based (16)	not (392), system (331), know (135), statement (126), think (111), context (105), difficult (93), definitely (86), correct (80), born (80), only (75), may (75), confused (75), incorrect (63), could (62), stated (62), don't (59), says (58), doesn't (57), **information** (54), states (53), no (53), first (52), **probably** (49), used (48), text (47)	film (346), american (296), first (194), known (188), (born (170), released (162), also (140), album (137), directed (127), united (120), based (120), new (116), born (109), football (106), one (100), – (94), band (91), best (89), played (88), written (86), south (81), world (79), city (77), series (77), population (77), name (77)
R2	film (301), american (266), known (166), (born (159), also (146), released (136), new (128), album (127), first (126), directed (114), one (112), series (110), united (97), – (95), television (95), band (87), state (86), based (83), written (82), **song** (79), national (76), played (74), best (69), located (67), city (66), football (66)	not (75), years (54), released (53), born (51), one (32), first (32), film (31), year (29), **ago.** (24), only (24), played (23), album (23), known (22), two (22), new (21), band (19), made (18), city (16), no (16), died (16), john (15), less (15), won (15), written (14), people (14), **lived** (14)	not (387), system (198), statement (125), know (93), doesn't (79), difficult (78), think (77), years (74), context (72), confused (70), may (65), only (63), born (61), states (60), correct (59), no (56), **ai** (55), definitely (55), released (52), text (50), incorrect (49), say (48), year (48), could (45), one (44), says (42)	film (332), american (280), released (189), known (188), (born (161), also (159), first (158), album (150), new (149), one (144), series (124), directed (123), band (106), united (105), television (101), not (98), played (97), – (97), written (96), state (96), song (89), born (88), based (87), national (83), city (82), located (80)
R3	not (197), one (146), said (122), new (116), would (104), first (92), some (91), make (87), people (83), may (83), also (80), **time** (77), no (75), – (75), **like** (74), **get** (74), last (72), only (68), two (68), **police** (66), made (61), think (55), **home** (54), **go** (54), **way** (53), **many** (53)	not (131), people (48), one (39), only (27), no (22), made (21), years (21), speaker (19), two (19), new (18), three (17), used (16), **use** (16), **person** (16), less (16), born (16), **good** (15), make (14), year (14), first (14), played (14), **school** (13), **government** (13), didn't (13), last (13), some (13)	not (527), statement (243), system (224), definitely (127), know (115), correct (104), says (98), no (91), doesn't (87), text (87), think (86), only (86), context (84), incorrect (81), may (81), **model** (75), could (74), confused (73), one (67), said (66), say (63), whether (58), difficult (57), neither (57), incorrect. (56), would (53)	not (328), one (185), new (134), people (131), said (127), would (115), first (106), some (104), make (101), no (97), may (95), only (95), two (87), time (86), last (85), like (83), get (82), made (82), also (80), – (75), **police** (74), use (67), many (66), three (63), **home** (62), **go** (62)
Contra.	american (219), film (216), new (146), (born (129), first (124), also (116), known (115), united (110), one (108), released (94), album (86), – (81), directed (78), series (76), may (72), best (71), television (70), band (69), not (68), based (66), written (65), south (65), national (63), two (62), **song** (60), football (59)	not (63), years (55), born (42), film (37), released (36), first (31), year (30), only (28), one (23), new (23), died (21), people (19), american (19), won (19), years. (19), **world** (18), three (18), played (18), album (17), two (17), less (17), directed (17), **old** (16), made (16), written (15), **lived** (15)	not (471), system (269), statement (174), incorrect (121), think (104), definitely (90), confused (87), difficult (83), only (78), born (71), says (63), context (61), years (57), states (51), one (50), would (49), incorrect. (47), know (42), name (42), **probably** (41), year (41), **ai** (41), could (40), first (38), may (38), model (35)	film (253), american (238), new (169), first (155), not (131), one (131), (born (130), released (130), known (126), also (125), united (119), album (103), directed (95), series (88), – (83), band (82), written (80), two (79), best (79), may (78), television (78), south (77), world (75), based (74), years (74), football (72)
Neut.	film (224), american (198), known (126), first (118), one (116), released (115), (born (112), also (107), album (101), new (97), not (95), directed (93), based (77), united (74), football (67), may (61), band (60), best (60), – (58), city (55), two (55), national (54), played (54), series (53), state (51), **song** (51)	not (63), one (37), born (36), released (29), only (28), never (25), played (24), film (22), people (21), made (19), first (18), no (18), new (17), album (17), won (17), known (16), population (15), john (14), two (14), last (14), name (13), united (13), died (12), best (12), football (11), written (11)	not (608), know (263), system (236), doesn't (157), no (150), context (147), statement (146), may (133), say (125), whether (124), correct (123), could (119), neither (117), don't (117), only (110), definitely (109), text (102), information (89), **nor** (83), mentioned (80), think (80), state (78), says (71), difficult (71), incorrect (69), confused (67)	film (246), american (208), not (158), one (153), released (144), known (142), first (136), album (118), new (114), (born (114), also (112), directed (101), united (87), based (83), played (78), football (78), only (76), best (72), band (70), two (69), made (69), city (66), may (64), born (63), name (63), written (60)
Entail.	film (207), american (171), known (136), first (134), also (132), (born (124), one (118), new (98), album (88), released (87), – (79), state (73), not (73), based (71), directed (69), series (67), united (65), played (61), written (61), best (60), television (60), **former** (60), two (58), band (56), may (55), **located** (53)	not (63), one (37), born (36), released (29), only (28), never (25), played (24), film (22), people (21), made (19), first (18), no (18), new (17), album (17), won (17), known (16), population (15), john (14), two (14), last (14), name (13), united (13), died (12), best (12), football (11), written (11)	not (608), know (263), system (236), doesn't (157), no (150), context (147), statement (146), may (133), say (125), whether (124), correct (123), could (119), neither (117), don't (117), only (110), definitely (109), text (102), information (89), nor (83), mentioned (80), think (80), state (78), says (71), difficult (71), incorrect (69), confused (67)	film (231), not (199), american (183), first (167), known (146), one (145), also (142), released (129), (born (124), new (116), album (103), born (91), state (84), years (82), two (81), based (81), – (79), directed (78), played (77), series (76), united (75), written (73), people (71), best (69), band (67), may (66)

Table 23: Top 25 most common words used by round and gold label. Bolded words are used preferentially in particular subsets.

Subset	Context	Statement	Rationale	Context+Statement+Rationale
ANLI	film (647), american (588), **known** (377), first (376), (born (365), **also** (355), one (342), new (341), released (296), album (275), **united** (249), **directed** (240), not (236), – (218), **based** (214), **series** (196), **best** (191), may (188), **band** (185), **state** (182), **football** (177), two (175), written (175), **television** (175), **national** (169), **south** (165)	not (252), born (132), years (120), released (107), one (87), film (83), first (82), only (76), **people** (75), year (61), **played** (58), new (58), two (54), **made** (54), album (49), no (46), **died** (46), **won** (46), **less** (44), **last** (42), american (41), **years.** (40), **three** (40), written (38), **used** (37), **john** (37)	not (1306), **system** (753), **statement** (494), **know** (343), **think** (274), **definitely** (268), **context** (261), **correct** (243), **difficult** (228), only (224), **doesn't** (223), may (221), **confused** (218), no (200), **says** (198), **incorrect** (193), **text** (184), **could** (181), **states** (166), born (160), one (155), **say** (147), years (146), **don't** (140), **would** (130), **whether** (129)	not (1794), film (802), system (781), american (659), one (584), first (563), statement (511), released (504), known (495), also (467), new (452), only (437), may (429), know (387), born (386), (born (371), album (362), no (337), think (337), based (335), years (332), two (313), states (313), united (308), state (304), directed (301)
Numerical	american (236), film (211), (born (162), first (151), known (138), album (136), new (129), released (126), also (117), united (117), one (109), – (101), band (87), series (83), best (82), television (79), directed (77), football (76), based (75), state (74), played (73), second (72), south (71), world (70), city (69), states (65)	years (114), born (79), released (74), **first** (61), year (52), not (44), **died** (38), **less** (37), two (36), one (35), years. (34), **three** (32), population (30), old (30), film (28), **ago.** (27), album (26), only (24), old. (24), **century.** (23), last (23), won (20), **least** (20), world (20), second (18), played (18)	not (344), system (291), statement (166), years (137), difficult (125), born (115), think (103), definitely (102), year (90), confused (90), only (88), correct (84), know (82), context (77), released (72), may (71), incorrect (70), first (61), text (60), could (59), would (57), one (55), says (51), doesn't (50), **mentioned** (49), died (48)	not (423), system (297), years (278), first (273), released (272), film (265), american (256), born (231), one (199), album (181), year (170), statement (169), (born (166), known (160), only (158), new (158), two (145), may (140), also (137), united (134), difficult (125), based (124), states (117), think (112), band (111), second (109)
Basic	film (238), american (193), one (143), known (138), new (135), first (134), also (132), not (125), released (105), directed (104), (born (100), album (99), state (97), united (90), may (83), song (80), based (78), series (74), best (74), two (73), television (72), – (69), south (68), written (68), said (65), would (64)	not (219), one (51), people (41), no (36), film (31), new (31), released (28), less (28), never (27), played (24), only (24), born (23), two (23), made (22), album (21), last (21), first (21), used (20), least (18), written (18), three (17), directed (17), best (16), years (16), **movie** (16), **good** (16)	not (546), system (290), statement (248), know (125), definitely (120), think (115), context (101), says (101), doesn't (97), correct (92), only (91), confused (89), may (88), incorrect (83), states (78), no (76), text (75), could (69), one (65), difficult (61), whether (58), would (58), say (56), neither (54), said (52), model (50)	not (890), system (303), film (298), one (259), statement (254), american (227), new (196), first (191), known (182), also (181), may (180), only (176), released (165), no (154), think (149), know (146), state (140), would (137), two (134), directed (133), album (132), states (128), based (127), says (127), people (126), said (123)
Reference	film (188), american (163), known (139), (born (128), also (112), first (98), one (85), new (83), directed (72), – (71), not (71), released (70), best (66), united (61), album (57), television (56), south (54), world (54), based (53), may (52), written (52), series (50), band (49),) (45), two (45), national (44)	not (70), born (39), years (33), name (23), film (21), made (20), won (19), one (19), people (19), first (19), only (17), year (17), played (16), released (16), died (16), known (15), band (15), speaker (14), new (14), written (14), three (13), two (12), no (12), man (12), directed (11), album (10)	not (358), system (199), statement (112), know (91), think (71), doesn't (70), confused (67), may (66), context (60), model (60), only (57), says (52), correct (52), could (51), definitely (50), name (50), difficult (49), born (46), one (42), probably (41), would (41), incorrect (40), states (39), don't (38), no (35), understand (34)	not (499), film (230), system (207), known (186), american (171), first (147), one (146), also (139), (born (129), may (126), statement (122), born (122), new (112), only (109), name (105), released (105), know (104), think (100), directed (93), years (89), would (88), written (84), two (83), states (82), based (82), best (80)
Tricky	film (227), american (142), first (110), known (104), one (102), also (99), new (93), (born (88), album (83), released (81), directed (77), based (75), song (71), not (68), series (65), written (61), united (60), band (59),) (55), may (51), – (50), south (48), only (48), two (48), television (46), located (44)	not (82), only (58), born (33), film (32), released (27), one (26), two (22), first (21), made (19), years (19), new (18), three (18), played (16), album (16), american (16), used (16), people (14), series (14), wrote (13), directed (13), written (13), also (13), band (13), known (13), won (13), **starts** (12)	not (386), system (204), statement (129), only (88), know (75), think (73), difficult (69), context (67), confused (66), incorrect (63), definitely (63), may (57), correct (54), says (51), states (49), doesn't (48), one (43), name (42), used (41), text (41), no (40), **ai** (38), don't (37), **words** (36), first (36), could (35)	not (536), film (281), system (208), only (194), one (171), first (167), american (166), also (146), known (141), statement (133), new (124), released (123), album (111), may (110), based (110), directed (99), two (92), (born (89), written (89), series (88), know (87), song (86), used (86), made (86), name (86), think (85)
Reasoning	film (390), american (363), (born (245), first (229), also (227), known (226), new (219), one (203), released (173), album (159), united (154), directed (151), not (147), based (138), – (125), football (124), state (117), national (116), played (111), best (110), band (109), television (108), may (108), series (106), former (105), south (104)	not (131), born (92), released (66), years (60), people (50), first (49), one (49), film (43), played (39), year (36), only (35), new (35), made (30), never (30), two (29), died (27), album (27), won (26), no (26), known (25), last (25), american (24), used (24), united (22), **john** (22), city (22)	not (919), system (466), know (291), statement (279), context (188), definitely (173), correct (172), doesn't (171), think (164), no (162), may (162), could (147), difficult (144), only (126), say (126), whether (123), says (119), confused (119), text (118), don't (114), neither (110), incorrect (110), born (101), one (96), information (95), states (92)	not (1197), system (483), film (481), american (411), one (348), first (335), know (312), released (307), known (306), also (292), new (290), statement (288), may (281), (born (250), only (249), born (249), no (239), state (218), based (213), album (206), think (200), played (196), united (196), context (191), could (184), doesn't (182)
Imperfections	film (87), american (76), also (54), one (52), first (47), known (45), released (45), new (44), album (42), not (36), based (35), directed (35), (born (35), city (34), united (33), written (31), two (30), song (29), – (26), series (25), band (25), people (25), television (24), population (24), name (24), national (24)	not (38), film (18), people (14), born (12), written (12), one (12), only (11), first (11), made (10), released (10), new (10), american (8), city (8), two (7), years (7), **popular** (7), many (6), different (6), united (6), album (6), **street** (6), show (6), also (6), population (6), three (6), **life** (5)	not (168), system (82), statement (70), know (50), correct (38), context (35), think (34), says (32), no (30), definitely (29), doesn't (28), confused (26), could (26), incorrect (26), one (24), states (23), only (23), stated (22), neither (22), may (21), model (21), say (21), text (20), don't (20), difficult (19), state (19)	not (242), film (116), american (94), system (89), one (88), statement (72), also (72), first (71), known (65), released (64), know (63), new (58), written (55), based (54), album (53), only (52), no (50), two (49), people (47), think (46), city (45), may (44), states (44), made (43), directed (42), united (42)

Table 24: Top 25 most common words used by annotation tag. Bolded words are used preferentially in particular subsets.

Concurrent hidden structure & grammar learning

Adeline Tan
University of California, Los Angeles
adelinertan@gmail.com

Abstract

The concurrent learning of both unseen structures and grammar is an enduring problem in phonological acquisition. The present study develops a joint model of word-UR-SR triples that incorporates a Maximum Entropy model of SRs conditioned on URs. The learner was presented with word-SR frequencies, and successfully learned the hidden structures and grammars that enabled it to generalize well on test data that were withheld during training. When given an option between acquiring a grammar that supported a rich base analysis and one that didn't, the learner always acquired the grammar that supported rich bases. These results suggest that the preference for acquiring a rich base grammar over a non rich base one is an emergent property of the proposed model.

1 Introduction

In order to fully acquire language, a child has to acquire both the representations and grammar of her language from observed surface forms. Representations include underlying forms, metrical structures, morphological boundaries within words, *etc.* Such representations are absent from the observed data that the child receives, and are thus termed hidden structure. The current study focuses on the learning of hidden structure(s) concurrently with the grammar.

Multiple approaches have been proposed for the concurrent learning of hidden structure and an accompanying constraint-based grammar (Tesar and Smolensky, 2000; Jarosz, 2015; Boersma and Pater, 2016; Rasin and Katzir, 2016; Nelson, 2019). Following Eisenstat (2009), Pater et al. (2012), Staubs and Pater (2016), Nazarov and Pater (2017), and O'Hara (2017), this study incorporates a Maximum Entropy (MaxEnt) grammar (Goldwater and Johnson, 2003) that governs the mapping between hidden structures and surface forms. The current study combines the word-hidden structure mapping with the hidden structure-surface form mapping by utilizing the chain rule of probability theory. This produces a joint word-underlying form-surface form (WORD-UR-SR) model that is compatible with a weighted-constraint grammar of UR-SR mappings. While the model is similar to the ones in Staubs and Pater (2016) and Nazarov and Pater (2017), the current study focuses on learning URs that an analyst would posit, with the learned grammar and lexicon subjected to generalization tasks with wug morphemes.

The model and the learner are introduced in §2 and §3 respectively. We then turn to several schematic languages, the first of which is based on English voicing assimilation (§4). This is followed by a set of six stress languages (§5). Two of the languages within the stress set allow multiple analyses, of which only one analysis supports rich bases (Prince and Smolensky, 2004), thus providing the opportunity to determine whether there is a preference for acquiring the rich base grammar over a non rich base one. The final schematic language is based on English velar softening (§6). §7 concludes.

2 Model

The knowledge whose acquisition will be investigated is knowledge of a particular distribution over WORD-UR-SR triples (*e.g.* <CROC-PL, /kɹɑk+z/, [kɹɑks]>: 99%; <CROC-PL, /kɹɑk+z/, [kɹɑkz]>: .003%; <CROC-PL, /kɹɑk+s/, [kɹɑks]>: .002%; . . .). In this paper, WORD[1] represents a sequence of morphemes, and morphemes are represented with uppercase letters.

[1] WORD is also abbreviated WD in this paper.

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 55-64.
Held on-line February 7-9, 2022

The probability of a triple can be rewritten as:

$$Pr(WD, UR, SR) = Pr(SR|WD, UR) \\ * Pr(WD, UR) \quad (1)$$

The first term, $Pr(SR|WD, UR)$, is the probability of an SR for a given WORD-UR pair, and is determined by the traditional phonological constraint grammar. For instance, if Pr([bæŋks]|BANK-PL, /bæŋk+z/) = 0.9, then we should interpret it to mean that the WORD-UR pair <BANK-PL, /bæŋk+z/> is realized as SR [bæŋks] 90% of the time. The model proposed here does not condition the UR-SR mapping on the word. Using the example above, this means that Pr([bæŋks]|BANK$_1$-PL, /bæŋk+z/) = Pr([bæŋks]|BANK$_2$-PL, /bæŋk+z/) = Pr([bæŋks]|BANK$_3$-3SG.PRES, /bæŋk+z/), where BANK$_1$ is the financial institution concept, BANK$_2$ is the river concept, and BANK$_3$ is the concept of turning at an angle. Consequently, $Pr(SR|WD, UR)$ = $Pr(SR|UR)$, and the probability of the WORD-UR-SR triple can be simplified to equation (2):

$$Pr(WD, UR, SR) = Pr(SR|UR) \\ * Pr(WD, UR) \quad (2)$$

Such probabilistic mappings of SRs conditioned on URs (*i.e.* $Pr(SR|UR)$) are computed by virtually all probabilistic constraint-based grammars (*e.g.* probabilistic OT, probabilitic versions of Harmonic Grammar, *etc.*) The current study uses a MaxEnt model, which is a weighted constraint grammar.

Following the traditional phonological MaxEnt model, each UR-SR pair (x, y) is associated with a feature vector, $\vec{v}(x, y)$, which captures the pair's properties. For UR-SR pairs, there are two classes of relevant properties. The first class concerns the form that the SR takes. For example, a feature may be used to track how many pairs of adjacent obstruents of an SR have different voicing values. Such features are known as markedness constraints. The second class of features concerns the mapping between the UR and the SR, and are most commonly used to penalize any changes between the two. These features are conventionally known as faithfulness constraints. Each feature has an associated weight, and the feature weights can be organized into the weight vector $\vec{w}$. The features of the UR-SR pair (x, y) are linearly combined (as in equation (3)[2]) to produce its harmony score, $h(x, y)$. $h(x, y)$ is essentially the weighted sum of the UR-SR pair (x, y)'s features, and is a scalar (rather than a vector).

$$h(x, y) = -(\vec{w} \cdot \vec{v}(x, y)) \quad (3)$$

The MaxEnt model then maps each pair's harmony score to its probability (equation (4)).

$$Pr(SR = y|UR = x) = \frac{e^{h(x,y)}}{Z(x)} \quad (4)$$

Since the traditional phonological MaxEnt grammar is a conditional ("discriminative") model, the partition function $Z(x)$ sums over all UR-SR pairs that share the same UR (equation 5).

$$Z(x) = \sum_{y' \in \mathcal{Y}_x} e^{h(x,y')} \quad (5)$$

In equation (5), $\mathcal{Y}_x$ is the set of all SRs that are compatible with UR x. This has the effect of normalizing the probability of a particular UR-SR mapping among only all other mappings from the same UR.

The second term in equation (2), $Pr(WD, UR)$, is the joint probability of a WORD-UR pair. This implicitly defines a conditional distribution $Pr(UR|WD)$ (equation (6)).

$$Pr(UR = x|WD = w) = \\ \frac{Pr(WD = w, UR = x)}{\sum_{x'} Pr(WD = w, UR = x')} \quad (6)$$

Under this conditional distribution we would expect $Pr(/\text{kɹɑk+z}/|\text{CROC-PL})$ to be high, and $Pr(/\text{kɹɑk+s}/|\text{CROC-PL})$, $Pr(/\text{kɹɑg+z}/|\text{CROC-PL})$, *etc.* to be low. For the morpheme CROC, the learner needs to choose between 2 possible stem-final segments: voiceless /k/ and voiced /g/[3]. For the plural morpheme, the learner needs to choose between voiceless /s/ and voiced /z/. Consequently, there are four potential URs that the learner considers for the word CROC-PL (Table 1). Table 1 also shows the four features for each

[2] A UR-SR pair is active for a phonological constraint when it violates the requirements of that constraint, which in turn reduces the pair's conditional probability. Hence the negative sign in equation (3).

[3] This example is modeled after English voicing assimilation where adjacent obstruents agree in voicing. The surface sequence [ks] could have arisen from any of the following UR sequences {/k+s/, /k+z/, /g+s/, /g+z/}. Hence, I vary only the stem-final segment, but none of the other stem segments.

of the four variants that the learner has to choose among. These features represent the strength of association between a particular morpheme and an aspect (*e.g.* morpheme-final obstruent voicing) of its UR. Within phonology, such features are also known as UR constraints (Zuraw, 2000; Boersma, 2001). Similar to its UR-SR counterpart, there is a feature vector $\vec{u}(w, x)$ for each WORD-UR pair (w, x). Likewise, the UR constraint weights can be organized into a vector $\vec{\theta}$. The harmony score for each WORD-UR pair is computed as per equation (7)[4].

$$g(x, y) = \vec{\theta} \cdot \vec{u}(x, y) \tag{7}$$

The harmony score of a WORD-UR pair is then mapped to its probability (equation 8).

$$Pr(WD = w, UR = x) = \frac{e^{g(w,x)}}{Z} \tag{8}$$

In contrast to the UR-SR model described above, the WORD-UR model is not conditional. The normalization takes place over all WORD-UR pairs (equation (9)).

$$Z = \sum_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}_w} e^{g(w,x)} \tag{9}$$

In equation (9), $\mathcal{W}$ is the set of words, and $\mathcal{X}_w$ is the set of all URs that are compatible with word w. This normalization produces a generative distribution over WORD-UR pairs, which in turn produces the generative distribution over WORD-UR-SR triples of equation (2). This departs from the models in Staubs and Pater (2016) and Nazarov and Pater (2017), which are discriminative models. A generative model is capable of describing differences in the frequencies of various words, in addition to the relationship between words and their realizations, whereas a discriminative model only does the latter.

3 Learning

The model takes in a set of WORD-SR pair frequencies (*e.g.* {<CROC-PL, [kɹɑks]>: 50; <CROC-PL, [kɹɑkz]>: 0; ...}), and learns a probability distribution over WORD-UR-SR triples (*e.g.* <CROC-PL, /kɹɑk+z/, [kɹɑks]>: 99%; <CROC-PL, /kɹɑk+z/, [kɹɑkz]>: .003%; <CROC-PL, /kɹɑk+s/, [kɹɑks]>: .002%; ...). The triple probability defined in Section 2 in fact implicitly defines a distribution over WORD-SR pairs as well. More concretely, the probability of pairs can be computed from the probability of triples via this summation:

$$\begin{aligned}
&Pr(WD = w, SR = y) \\
&= \sum_x Pr(WD = w, UR = x, SR = y) \\
&= \sum_x Pr(SR = y | UR = x) * Pr(WD = w, UR = x) \\
&= \sum_x \frac{e^{h(x,y)}}{Z(x)} * \frac{e^{g(w,x)}}{Z}
\end{aligned} \tag{10}$$

The likelihood $Pr(WD, SR)$ can be understood as a function of the parameters $\vec{w}$ and $\vec{\theta}$. Experimentation showed that regularization terms did not improve performance in fitting to test data that was withheld from training, so the learner's objective is to seek the values of $\vec{w}$ and $\vec{\theta}$ that maximize this likelihood. In order to assess which values of $\vec{w}$ and $\vec{\theta}$ will be found by the learner, I use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). Notice that $Pr(WD, SR)$ is a marginal distribution[5]. The likelihood function of marginal distributions is not guaranteed to be convex, so each EM run finds a local maximum. I take the highest of these local maxima to identify the predicted outcome of learning.

4 English Voicing Assimilation

In English voicing assimilation, adjacent obstruents with different voicing values are resolved with suffixes assimilating their voicing value to that of the stem. The underlying voicing value of stem-final obstruents and suffixes constitute the hidden structures.

4.1 Experimental setup

The first language had the words {CROC-PL, DOG-PL, COW-PL}. Its grammar had the constraints {AGREE(voice), IDENT$_{\text{stem}}$, IDENT$_{\text{general}}$}, so $\vec{w}$ was 3-dimensional for this language. In addition, six potential UR variants {/kɹɑk/, /kɹɑg/, /dɑk/, /dɑg/, /-s/, /-z/} were considered, making $\vec{\theta}$ 6-dimensional. These nine dimensions correspond to the first nine rows in Table 3.

What constitutes successful learning? First, we can check whether the UR learned for each morpheme of the training data matches what would

[4] A WORD-UR pair is active for a particular UR constraint when it contains the morpheme, segment, *etc.*, required by that constraint, which in turn increases the pair's probability. Hence the sign difference between equations (3) and (7).

[5] The summation in equation (10) produces marginal probabilities.

WORD	UR$_{\text{WORD}}$	(CROC, /kɹɑk/)	(CROC, /kɹɑg/)	(PL, /-s/)	(PL, /-z/)
CROC-PL	/kɹɑk+s/	1	0	1	0
	/kɹɑg+s/	0	1	1	0
	/kɹɑk+z/	1	0	0	1
	/kɹɑg+z/	0	1	0	1

Table 1: UR constraints for the word CROC-PL.

be predicted via traditional phonological analyses. For example, a phonologist would posit that a child learns the URs /kɹɑk/, /dɑg/ and /-z/ for the CROC, DOG and -PL morphemes respectively. Recall that the model produces a distribution over WORD-UR-SR triples. In order to find the probability of the word-sized UR containing /-z/ in the appropriate position, given that the WORD (*i.e.* sequence of morphemes) has -PL in that position, we apply the following equation:

$$p(\text{/-z/}|\text{-PL}) = \frac{\begin{array}{c}[Pr(\text{UR=/dɑk-z/, WD=DOG-PL}) \\ +Pr(\text{UR=/dɑg-z/, WD=DOG-PL}) \\ +Pr(\text{UR=/kɹɑk-z/, WD=CROC-PL}) \\ +Pr(\text{UR=/kɹɑg-z/, WD=CROC-PL}) \\ +Pr(\text{UR=/kaw-z/, WD=COW-PL})]\end{array}}{\begin{array}{c}[Pr(\text{WD=DOG-PL})+Pr(\text{WD=CROC-PL}) \\ +Pr(\text{WD=COW-PL})]\end{array}} \quad (11)$$

The probability of a particular UR for each morpheme is calculated in the same manner for each of the other morpheme-UR pairs.

Second, the model must be able to generalize to unseen data in the way that humans do. Unseen data are WORD-SR pairs that the model wasn't provided with in the training set. For English voicing assimilation, the words in Table 2 provide a good test set for generalizability. Each test word is composed of a new morpheme and an old morpheme. This isolates each of the old morphemes, so that only one old morpheme appears in each word. This allows us to test what the model knows between /-s/ *vs.* /-z/ as the UR of the -PL morpheme, as well as what the model knows about how /-s/ & /-z/ are realized on the surface. The new nouns WUG /wʌg/, HEAK /hik/ & CRA /kɹɑ/ will illuminate what the model had learned about the -PL suffix. I introduce a novel suffix with UR /-d/ (representing some morpheme I will write as -D), to test what the model had learned about the roots DOG and CROC. For example, if Pr((/hik-z/, [hiks])|HEAK-PL) is high, then we'd know that the model generalizes in the same way that English speakers do, as evidenced by wug tests. The probability of a UR-SR pair for a given word is calculated as per equation (12).

WORD
WUG-PL
HEAK-PL
CRA-PL
DOG-D
CROC-D

Table 2: Test set words for English voicing assimilation.

$$Pr(UR = x, SR = y|WD = w) = \frac{Pr(WD = w, UR = x, SR = y)}{\sum_{x'} \sum_{y'} Pr(WD = w, UR = x', SR = y')} \quad (12)$$

4.2 Results

The training data consisted of 10 logically possible WORD-SR pairs, of which only three were observed. Each of the three observed pairs {(CROC-PL, [kɹɑks]), (DOG-PL, [dɑgz]), (COW-PL, [kawz])} was only observed once. The learner sought the parameter values that maximized the likelihood of the training data. Five settings of the parameters are shown in Table 3. I found these by running the EM algorithm from 20 randomly initialized[6] starting points. The likelihood of the training data for each of the five parameter settings is $0.33 \times 0.33 \times 0.33 = 0.33^3$. These five settings have already hit the maximum likelihood of training data – there isn't another parameter setting that would provide a much better likelihood since these settings have already matched the empirical relative frequencies almost perfectly.

[6] Initial weights for all eight languages in the present study were drawn from a uniform distribution with range=[0.1, 5) for phonological constraints & range=[0, 100) for UR constraints.

[7] In this paper, negative weights were only allowed for UR constraints. Weights for regular phonological constraints were not allowed to be negative. For voicing assimilation, the UR constraint (COW, /kaw/) was excluded from the set of features, since the morpheme COW had only one underlying form under consideration. This resulted in (DOG, /dɑg/) attaining 0 weight, which pushed (DOG, /dɑk/) to a negative weight. Because it is the difference between weights rather than the actual value of the weights that matter, the negative weights do not have any meaningful impact on the results.

	1	2	3	4	5
AGREE(voice)	24.7	40.9	24.4	33.3	26.5
IDENT$_{\text{stem}}$	15.0	11.9	12.6	29.4	11.6
IDENT$_{\text{general}}$	11.9	18.5	12.0	18.5	13.6
(DOG, /dɑk/)	-43.2	-11.9	-30.0	-26.4	-11.6
(DOG, /dɑg/)	0.0	0.0	0.0	0.0	0.0
(CROC, /kɹɑk/)	0.0	0.0	0.0	0.0	0.0
(CROC, /kɹɑg/)	-16.3	-23.7	-20.5	-33.1	-13.3
(-PL, /-s/)	0.3	14.4	9.5	20.7	24.6
(-PL, /-z/)	91.0	56.8	76.6	83.0	77.4
Pr(CROC-PL, [kɹɑks])	0.33	0.33	0.33	0.33	0.33
Pr(DOG-PL, [dɑgz])	0.33	0.33	0.33	0.33	0.33
Pr(COW-PL, [kawz])	0.33	0.33	0.33	0.33	0.33
Likelihood of training data	0.33^3	0.33^3	0.33^3	0.33^3	0.33^3
Negative log-likelihood of training data	-3.29585	-3.29585	-3.29585	-3.29584	-3.29586

Table 3: Feature weights[7], probability of observed data, & likelihood of training data from the best five runs (English voicing assimilation).

Recall the first criterion of successful learning: the learner has to learn the very same morpheme-sized URs that human learners are posited by phonologists to learn. Applying the equation in (11), we see that the lexicon learned is indeed the one that matches with traditional phonological analysis (Table 4[8]). A further examination of the UR constraints confirms that the UR constraints associated with expected URs (*i.e.* DOG has underlying root-final /g/, CROC has underlying root-final /k/, the plural morpheme is underlying /-z/) have higher weights than their counterparts (Table 3).

	1	2	3	4	5
p(/dɑg/|DOG)	1.0	1.0	1.0	1.0	1.0
p(/kɹɑk/|CROC)	1.0	1.0	1.0	1.0	1.0
p(/-z/|-PL)	1.0	1.0	1.0	1.0	1.0

Table 4: Lexicon (English voicing assimilation).

To fulfill the second criterion of successful learning, the learned models had to generalize in the same way that English speakers generalize. The generalization task considered 8 UR-SR combinations for each word[9]. For nonce word HEAK /hik/, an English speaker would produce [hiks] for HEAK-PL via the UR /hik-z/. Thus, successful generalization for the word HEAK-PL required assigning high probability to the UR-SR pair (/hik-z/, [hiks]), and low probability to the seven other pairs. Table 5 presents, in the top five rows, $Pr((\text{UR, SR})|\text{WD})$ for the UR-SR pairs that are expected to have high probability for their respective words, given what we know about how English speakers behave on wug tests. The results in Table 5[10] indicate that all five models very successfully generalized in a manner that mimicked speakers, with probabilities close to or at 100%. A look at the learned phonological constraint weights in Table 3 shows why all five parameter settings mirrored speakers so well in the generalization task. The models all learned the two crucial weight-inequalities required for English voicing assimilation: AGREE(voice) $>$ IDENT$_{\text{general}}$ as well as IDENT$_{\text{stem}} > 0$.

For voicing assimilation, the model did well on both the UR-learning of morphemes & wug-test mirroring tasks because both the lexicon & grammar learned by the learner mirrored what English speakers are believed to have learned for voicing assimilation.

5 PAKA stress languages

5.1 Experimental setup

The next 6 languages were generated using similar morphemes and constraints that generated the PAKA World dataset in Tesar et al. (2003). There were two roots: (PA & BA), as well as two suffixes (-KA & -GA). The URs of PA and KA were always unstressed: /pa/ & /-ka/. In contrast, the

[8] All probabilities in Table 4 were very close to or at 100%. The lowest was 99.9991% for p(/dɑg/|DOG) of the fifth parameter setting.

[9] To illustrate the 8 combinations, consider the nonce word WUGS. 2 variations are available via the UR: /-s, -z/, 2 via the SR of the stem-final consonant: [wʌk, wʌg], and 2 via the SR of the suffix consonant: [-s, -z]. This produces $2^3 = 8$ UR-SR combinations.

[10] All probabilities in Table 5 were very close to or at 100%. The lowest was 99.9986% for Pr((/dɑg-d/, [dɑgd])|DOG-D) of the fifth parameter setting.

	1	2	3	4	5
Pr((/wʌg-z/,[wʌgz])$\mid$WUG-PL)	1.0	1.0	1.0	1.0	1.0
Pr((/hik-z/,[hiks])$\mid$HEAK-PL)	1.0	1.0	1.0	1.0	1.0
Pr((/kɹɑ-z/,[kɹɑz])$\mid$CRA-PL)	1.0	1.0	1.0	1.0	1.0
Pr((/dɑg-d/, [dɑgd])$\mid$DOG-D)	1.0	1.0	1.0	1.0	1.0
Pr((/kɹɑk-d/, [kɹɑkt])$\mid$CROC-D)	1.0	1.0	1.0	1.0	1.0
Pr((/wʌg-s/,[wʌgz])$\mid$WUG-PL)	0.0	0.0	0.0	0.0	0.0
Pr((/wʌg-z/,[wʌkz])$\mid$WUG-PL)	0.0	0.0	0.0	0.0	0.0
⋮	⋮	⋮	⋮	⋮	⋮

Table 5: Probability of UR-SR pair for a given test set word (English voicing assimilation).

URs of BA & GA could bear stress, so the model considered the potential-URs $\{$/ˈba/, /ba/, /-ˈga/, /-ga/$\}$. Accordingly, the relevant UR constraints for this dataset were: (BA, /ˈba/), (BA, /ba/), (GA, /-ˈga/) & (GA, /-ga/). The four features that went into $Pr(SR|UR)$ were:

- MAINLEFT (ML)
  - Stress the leftmost syllable.
- MAINRIGHT (MR)
  - Stress the rightmost syllable.
- MAX$_{\text{general}}$-STRESS (F)
  - If a syllable is stressed in the UR, retain its stress in the SR.
- MAX$_{\text{root}}$-STRESS (FR)
  - If a root syllable is stressed in the UR, retain its stress in the SR.

These morphemes & constraints produced six logically possible languages[11], which are the six sets of observed SRs shown in Table 6. Languages 3, 4 & 6 were each compatible with only 1 lexicon. Language 5 was compatible with both /-ga/ & /-ˈga/, but compatible with only a single grammar. For Languages 1 & 2, four lexicon-grammar combinations were available for each language.

5.2 Results

For each language, the training data had 12 logically possible WORD-SR pairs[13], of which four pairs were each observed once. The four SRs for each language can be read off column "Observed SRs" of Table 6. As with English voicing assimilation, I did 20 EM runs per language. For all six languages, the learner succeeded in finding multiple parameter settings that hit the maximum likelihood of training data $0.25 \times 0.25 \times 0.25 \times 0.25 = 0.25^4$. For the sake of brevity, only one of these parameter settings is presented for each of the six languages (Table 7).

To test generalizability, two new roots $\{$SO /so/, ZE /ˈze/$\}$ and two new suffixes $\{$-FO /-fo/, -VE /-ˈve/$\}$ were introduced to form test set words (Table 8). As expected, all parameter settings that attained the maximum likelihood of training data generalized to test words at near 100% probability. A sample of the probabilities of UR-SR pairs for test word BA-FO is shown for one simulation of Language 3, where the combination of trained morpheme BA with an unaccented suffix like /-ka/ produced stress on the first syllable. Likewise, when BA combines with unstressed /-fo/, successful generalization requires a UR-SR pair with [ˈbafo] to have high probability (Table 9).

5.3 Rich base supporting grammars

According to Prince and Smolensky (2004), the role of a constraint-based grammar is to assign an output to each input[15]. In the case of absolute ill-formedness (*e.g.* absence of right-stressed SRs in a left-stressed language), the grammar (*i.e.* the constraint interactions that govern the UR-SR mapping) must ensure that no input ever leads to ill-formed outputs (*e.g.* not even a UR with rightmost

[11] Since MaxEnt generates probabilistic languages, there are technically an infinite number of possible languages. However, I'm restricting the set of languages to only those where there is effectively only one winning SR per UR.

[12] Since the model is MaxEnt rather than non-probabilistic Harmonic Grammar, the difference between the terms on both sides of an inequality need to be sufficiently large in order to generate categorical outcomes. Determining exactly how large a difference is needed for each inequality is difficult. Nevertheless, the test task provides a way to check that the trained weights indeed produce sufficient difference between the two terms of an inequality. If the difference were not sufficiently large, the test task would fail to produce categorical outcomes.

[13] There were three SRs per word – left-stress, right-stress, and no stress at all.

[14] UR constraints for the morphemes PA & -KA that do not have multiple URs under consideration were included in this feature set. Hence negative UR constraint weights do not make an appearance here.

[15] Prince and Smolensky (2004) were writing about Optimality Theory, where the grammar consisted of ranked constraints picking a sole output for each input. Nevertheless, the grammar's role in mapping inputs to outputs still holds for probabilistic constraint-based grammars.

Lg	*Observed SRs*	Description	*Lexicon*	*Required weight inequalities*[12]
1	[ˈpaka, ˈpaga, ˈbaka, ˈbaga]	predictable left-stress	/ba, -ga/ /ˈba, -ga/	ML > MR
			/ba, -ˈga/ /ˈba, -ˈga/	ML > MR + F
2	[paˈka, paˈga, baˈka, baˈga]	predictable right-stress	/ba, -ga/ /ba, -ˈga/	MR > ML
			/ˈba, -ga/ /ˈba, -ˈga/	MR > ML + F + FR
3	[ˈpaka, paˈga, ˈbaka, ˈbaga]	full accentual contrast, default left	/ˈba, -ˈga/	MR + F > ML > MR
4	[paˈka, paˈga, ˈbaka, baˈga]	full accentual contrast, default right	/ˈba, -ˈga/	ML + F + FR > MR > ML + FR
5	[paˈka, paˈga, ˈbaka, ˈbaga]	contrast in roots only, default right	/ˈba, -ga/ /ˈba, -ˈga/	ML + FR > MR > ML
6	[ˈpaka, paˈga, ˈbaka, baˈga]	contrast in suffixes only, default left	/ba, -ˈga/	MR + F > ML > MR

Table 6: PAKA languages and respective logically-possible lexicon-grammar combinations.

	Lg 1	*Lg 2*	*Lg 3*	*Lg 4*	*Lg 5*	*Lg 6*
MAINLEFT (ML)	19.6	0.0	16.9	8.4	0.0	44.7
MAINRIGHT (MR)	0.0	20.0	0.0	23.3	19.4	0.0
$\text{MAX}_{\text{general}}$-STRESS (F)	1.5	1.7	32.4	37.4	21.7	111.4
MAX_{root}-STRESS (FR)	3.1	0.0	7.5	0.0	23.2	2.2
(BA, /ba/)	69.9	47.3	15.5	36.4	27.8	81.1
(BA, /ˈba/)	0.5	77.2	61.4	80.5	66.7	34.2
(GA, /-ga/)	59.6	28.1	11.1	25.8	71.4	59.3
(GA, /-ˈga/)	6.2	65.2	61.2	85.0	35.2	76.8
Likelihood $_{\textit{training data}}$	0.25^4	0.25^4	0.25^4	0.25^4	0.25^4	0.25^4
Negative log-likelihood $_{\textit{training data}}$	-5.545177	-5.545177	-5.545178	-5.545178	-5.545177	-5.545177

Table 7: Feature weights[14] & likelihood of training data from the best runs for each PAKA language.

WORD
SO -FO
SO -GA
BA -FO
ZE -GA
BA -VE

Table 8: Test set words for the six PAKA languages.

WORD	UR-SR pair	Pr(UR, SR|WD)
BA -FO	/ba-fo/, [bafo]	2.7×10^{-12}
	/ba-fo/, [ˈbafo]	4.6×10^{-5}
	/ba-fo/, [baˈfo]	2.7×10^{-11}
	/ˈba-fo/, [bafo]	1.8×10^{-21}
	/ˈba-fo/, [ˈbafo]	9.9995×10
	/ˈba-fo/, [baˈfo]	1.7×10^{-20}

Table 9: Generalization to BA-FO in Language 3 (one run shown).

stress can produce an SR with rightmost stress). Within models that feature probabilistic UR-SR mappings, this translates to the grammar ensuring that no inputs ever map to ill-formed outputs with anything other than a vanishingly small probability. In other words, the grammar should be fail-safe; it should be able to map all URs (even implausible ones like a right-stressed UR in a left-stressed language) to SRs with appropriate probability values. This concept is known as the Richness of the Base (Prince and Smolensky, 2004).

Language 1 & Language 2 are languages with predictable left- and right-stress respectively. Each of these two languages is compatible with two grammars (Table 6). In Language 1, the two possible grammars are Grammar 1 ($ML > MR$) & Grammar 2 ($ML > MR + F$). The lexicon that includes /-ga/ minimally requires Grammar 1, while the lexicon that includes /-ˈga/ minimally requires Grammar 2. Since the weights of phonological constraints could not be negative, Grammar 2 entails Grammar 1. It follows that Grammar 2 is compatible with both /-ga/ & /-ˈga/ while Grammar 1 is compatible with only /-ga/. Grammar 2 is thus a grammar that supports rich bases because it is capable of producing SRs with the right probabilities even with an implausible UR (underlying stressed suffix /-ˈga/ in a left-stressed language where unstressed root /pa/ also exists). In contrast, Grammar 1 is the less restrictive grammar because it requires only $ML > MR$, thus allowing $MR + F > ML$. The gang effect of right-stress (MR) and general faithfulness (F) over left-stress (ML) results in /pa-ˈga/ surfacing with rightmost stress *[pa-ˈga]. For Language 2, this entailment relation also holds amongst its two grammars, with Grammar 4 ($MR > ML + F + FR$) being the rich base supporting grammar, and Grammar 3 ($MR > ML$) the less restrictive one. These two languages thus provide useful test cases on whether there is a preference for a rich base supporting grammar over its less restrictive rival or vice versa.

All 20 runs for Languages 1 & 2 always learned the rich base grammar. Trained weights for an example run of Language 1 is shown in Table 7, where the rich base grammar, ML (19.6) $> MR + F$ (0 + 1.5), is learned. To test this further, I ran 200 more simulations for both languages. All 200 runs for both languages always learned the rich base supporting grammar, sometimes with the lexicon that minimally required the rich base grammar, and sometimes with the lexicon that minimally required the less restrictive one. This indicated a strong preference for learning the rich base grammar over its less restrictive counterpart.

The preference for acquiring the rich base grammar is an emergent property of the model. EM finds the local maximum by hill-climbing from a randomly initialized point within the solution space. The solution space is the likelihood function of the marginal distribution (equation (10)) of the model defined in §2. Hill-climbing (*i.e.* gradient ascent) is guided by the gradients of the solution space at the current point. The preference for converging at maxima corresponding to the rich base grammar indicates the following: within the solution space, there are more points with gradients pointing towards maxima corresponding to the rich base grammar and fewer points with gradients pointing towards maxima corresponding to the less restrictive grammar. Since the solution space is a property of the model (rather than that of a particular learner), models with similar architecture (*e.g.* Staubs and Pater (2016); Nazarov and Pater (2017)) are likely to also favor the acquisition of rich base grammars.

6 Velar Softening

In English velar softening, /k/ → [s] before a high front vowel when a morpheme-boundary intervenes (*e.g.* *electri*[k]∼*electri*[s]-*ity*). Velar softening is an instance of the derived environment effect (DEE) because its triggering environment requires the presence of a morpheme boundary. DEEs are a puzzle because both the alternation and the segmentation into morphemes must be acquired simultaneously. In an additional wrinkle, DEEs are often only triggered by specific morphemes. For example, the *-ity* morpheme triggers velar softening, but the *-ish* morpheme does not. Velar softening thus has three sources of hidden structure – presence of a morpheme boundary, whether a particular suffix is exceptional in triggering velar softening, and the usual UR-segment-learning (/k/ or /s/) that we've already seen in the preceding test cases. I use the *-symbol to indicate the exception tagged UR variant.

6.1 Experimental setup

There were eight observed words {ELECTRIC, ELECTRICITY, ELECTRICISH, KITTY, SECURE, SECURITY, SMALL, SMALLISH}. The three sources of hidden structure were combined to produce URs for these eight words. The URs for ELECTRICITY are shown with their relevant UR constraint features (Table 10). The UR /electrik-*ity/ contained the morphemes ELECTRIC[16] & -ITY, so it was active for those two features. These features represent a new class of UR constraint, namely those that indicate the presence of certain morphemes. Such features were not required in the preceding cases as the morpheme boundary was not in question. The UR /electrik-*ity/ had underlying /k/ for morpheme ELECTRIC, and the exception-tagged version of the -ITY suffix, so it was also active for features (ELECTRIC, k)[17] & (-ITY, -*ity) respectively. These UR constrains are the same kind that we've seen before.

Three phonological constraints controlled the UR-SR mapping – a general markedness constraint against [kɪ] sequences (M), an exception-tagged version that was active only when an exception-tagged morpheme was part of the [kɪ] sequence (ME), and a general IDENT constraint (F_{gen}).

[16] Abbreviated as EL...C in Table 10.

[17] Abbreviated as (EL...C, k) in Table 10.

UR$_{\textbf{ELECTRICITY}}$	EL...C	EL...CITY	-ITY	(EL...C, k)	(EL...C, s)	(EL...CITY, k)	(EL...CITY, s)	(-ITY, -*ity)	(-ITY, -ity)
/electrik-*ity/	1	0	1	1	0	0	0	1	0
/electrik-ity/	1	0	1	1	0	0	0	0	1
/electris-*ity/	1	0	1	0	1	0	0	1	0
/electris-ity/	1	0	1	0	1	0	0	0	1
/electrikity/	0	1	0	0	0	1	0	0	0
/electrisity/	0	1	0	0	0	0	1	0	0

Table 10: URs under consideration for the word ELECTRICITY shown with their UR constraint features (English velar softening).

6.2 Results

The training data had 12 logically possible WORD-SR pairs, of which eight were each observed once. Of 125 EM runs, multiple parameter settings were found to have reached the maximum likelihood of training data $0.125^8 = 5.9605 \times 10^{-5}$. 90.9% of these parameter settings learned the very same hidden structures that matched the standard phonological analysis of velar softening. This included learning that the -ITY morpheme was exception-tagged but that -ISH wasn't. All of these hidden features were learned at probabilities[18] greater than 97%, with the lowest going to the probability of a morpheme boundary in SECURITY at 97.3%.

To test whether the learned grammars generalized in a way that mimicked human learners, the following three morphemes were introduced: a new root with a morpheme-final-/k/ (CLEMIC /ˈklɛmɪk/), an exception-tagged suffix *-ISM /*-ɪzm/ and a non-exception-tagged suffix -Y /-i/ to create the test words in Table 11. For each test word, its expected UR-SR pair arose from what traditional phonological analysis would have a child positing as its UR and SR. For instance, the word ELECTRIC-ISM would be posited to have morpheme-final /k/ as opposed to /s/ for ELECTRIC, with that /k/ surfacing as [s]. The expected UR-SR pair for each word is shown in column "Expected UR-SR" of Table 11. The same 90.9% of parameter settings that hit the maximum likelihood of training data generalized well to the test set with probabilities of the expected UR-SR pair for each word approaching 100%[19]. Examination of the weights learned for the parameter settings that successfully generalized confirmed that they had each learned the grammar necessary for velar softening. That is, the alternation applied when the suffix was exception-tagged ($ME + M > F_{gen}$), but did not take place when the suffix wasn't exception-tagged ($F_{gen} > M$).

WORD	Expected UR-SR
ELECTRIC-ISM	electri/k/-*ism, electri[s]ism
ELECTRIC-Y	electri/k/-y, electri[k]y
CLEMIC-ITY	clemi/k/-*ity, clemi[s]ity
CLEMIC-ISH	clemi/k/-ish, clemi[k]ish

Table 11: Test set words for English velar softening & UR-SR pairs expected to be learned by human learners.

What would this high-but-not-100% rate of acquisition of the velar softening grammar mean for human learners? Perhaps 10% of people fail to learn the velar softening grammar, and instead rely on memorized forms for existing words. These people are predicted to not apply velar softening in a wug test. Interestingly, in a wug test with nonce stems and the *-ity* suffix, Pierrehumbert (2006) found that 2 in 10 subjects did not have productive velar softening.

7 Conclusion

The present study produced a domain-general model that concurrently learned both hidden structure and a weighted-constraint grammar. The model was trained on eight languages, and generalized well to test data on all of them. Two languages in particular presented a choice between acquiring a grammar that supported rich bases versus one that didn't. This study found a strong preference for acquiring the rich base grammar, which I argued was an emergent property of the model. The present study thus presented a way in which a rich base grammar may be acquired when URs are not known in advance.

Acknowledgments

I would like to thank Tim Hunter and Kie Zuraw for stimulating discussion which made this paper better. I also thank the three anonymous reviewers for helpful feedback.

[18] These probability values were calculated using the same method shown in eq (11).

[19] The lowest probability was $Pr(/\text{klɛmɪk-ɪʃ}/, [\text{klɛmɪkɪʃ}] | \text{CLEMIC-ISH}) = 99.983\%$.

References

Paul Boersma. 2001. Phonology-semantics interaction in OT, and its acquisition. In Robert Kirchner, Wolf Wikeley, and Joe Pater, editors, *Papers in Experimental and Theoretical Linguistics*, volume 6, pages 24–35. University of Alberta, Edmonton.

Paul Boersma and Joe Pater. 2016. Convergence properties of a gradual learning algorithm for harmonic grammar. In John J. McCarthy and Joe Pater, editors, *Harmonic Grammar and Harmonic Serialism*. Equinox Press, London.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B (Methodological), 39(1):1–38.

Sarah Eisenstat. 2009. Learning underlying forms with maxent. Master's thesis, Brown University.

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In Jennifer Spenader, Anders Eriksson, and Östen Dahl, editors, *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, pages 111–20.

Gaja Jarosz. 2015. Expectation driven learning of phonology, ms. University of Massachusetts, Amherst.

Aleksei Nazarov and Joe Pater. 2017. Learning opacity in stratal maximum entropy grammar. *Phonology*, 34:299–324.

Max Nelson. 2019. Segmentation and UR acquisition with UR constraints. *Proceedings of the Society for Computation in Linguistics*, 2:60–68.

Charlie O'Hara. 2017. How abstract is more abstract? learning abstract underlying representations. *Phonology*, 34:325–345.

Joe Pater, Robert Staubs, Karen Jesney, and Brian Smith. 2012. Learning probabilites over underlying representations. *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 62–71.

Janet Pierrehumbert. 2006. The statistical basis of an unnatural alternation. In Louis Goldstein, D. H. Whalen, and Catherine T. Best, editors, *Laboratory Phonology VIII, Varieties of Phonological Competence*, pages 81–107. Mouton de Gruyter, Berlin.

Alan Prince and Paul Smolensky. 2004. *Optimality Theory: Constraint interaction in generative grammar*. Blackwell Publishing, Malden, MA.

Ezer Rasin and Roni Katzir. 2016. On evaluation metrics in Optimality Theory. *Linguistic Inquiry*, 47(2):235–282.

Robert Staubs and Joe Pater. 2016. Learning serial constraint-based grammars. In John McCarthy and Joe Pater, editors, *Harmonic Grammar and Harmonic Serialism*. Equinox Press.

Bruce Tesar, John Alderete, Graham Horwood, Nazarré Merchant, Koichi Nishitani, and Alan Prince. 2003. Surgery in language learning. In G. Garding and M. Tsujimura, editors, *WCCFL 22 Proceedings*, pages 477–90. Cascadilla Press, Somerville, MA.

Bruce Tesar and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press, Cambridge, MA.

Kie Zuraw. 2000. *Patterned exceptions in phonology*. Ph.D. thesis, UCLA.

Evaluating Structural Economy Claims in Relative Clause Attachment

Aniello De Santo
Department of Linguistics
University of Utah
aniello.desanto@utah.edu

So Young Lee
Department of English
Miami University
lees44@miamioh.edu

Abstract

Grillo and Costa (2014) argue for a pseudo-relative (PR) first account of relative clause attachment preferences (RC) such that, when faced with a sentence ambiguous between a PR and a RC interpretation, the parser prefers committing to a PR structure first, thus giving rise to what looks like a high-attachment preference. One possible explanation for this parsing choice is in terms of simplicity of the PR structure, and overall economy principles. Here, we evaluate this hypothesis by testing the predictions of a parser for Minimalist grammars for PR and RC structures in Italian. We discuss the relevance of our results for PR-first explanations of the cross-linguistic variability of RC attachment biases, and highlight the role that computational models can play in evaluating the cognitive plausibility of economy considerations tied to fine-grained structural analyses.

1 Introduction

The idea that economy and simplicity principles affect syntactic derivations has been central to inquiries in Generative grammar. In earlier iterations, economy conditions were basically conceived as evaluation metrics for selecting grammars from the format permitted for rule systems. In a lexicalized, non-rule based framework such as the Minimalist Program, economy has come to play a different but still central guiding role in the theoretical architecture of the grammar — motivating, for instance, the preference for applying some grammatical operations over others (Chomsky, 1995; Collins, 2001).

These appeals to economy considerations, ubiquitous even in the most recent syntactic literature, occasionally reference general *parsing* and computational motivations (Kayne, 1994; Motut, 2010; Fukuda, 2011; Razaghi et al., 2015; Bošković and Messick, 2017, a.o.). In this sense, a mutual exchange of questions and insides across the syntactic and psycholinguistic literature has been fruitful, inspiring a vast array of research questions. For instance, there are numerous detailed formalizations of the role of *locality* considerations in our understanding of grammatical and processing principles (Frazier, 1987, 1978; De Vincenzi, 1991; Gibson, 2000). However, within the theoretical literature there is sometime the tendency to rely on economy explanations without overtly specifying what kind of assumptions are made about fine-grained syntactic details and their relation to broader principles of *cost*. For example, while it is possible to find many claims of *structural simplicity* made to motivate syntactic and/or psycholinguistic predictions, it is often unclear in these contexts how *simplicity* is actually quantified, how these computational demands would be implemented in a precise parsing architecture, and how these costs are linked to cognitive resources. Ideally, it would be desirable to formally spell-out the kind of complexity assumptions underlying different aspects of syntactic representations, so to explore the plausibility of the predictions made by economy claims with respect to behavioral responses in psycholinguistic studies (Bresnan, 1978, 1982; Rambow and Joshi, 1994; Kobele et al., 2013; Demberg and Keller, 2009).

Following these intuitions, in this paper we focus on economy principles as referenced in the context of the cross-linguistic variation of relative clause attachment ambiguity preference. We suggest that a transparently specified computational model which takes syntactic assumptions seriously can help shed light on these issues. In particular, we propose the use of a parser for Minimalist grammars (MGs; Stabler, 2013), coupled with complexity metrics measuring memory usage (Kobele et al., 2013; Gerth, 2015; Graf et al., 2017, a.o.), in order to investigate the predictions of the so-called *pseudo-relative first* hypothesis (Grillo and Costa, 2014) in a framework that actually formalizes

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 65-75.
Held on-line February 7-9, 2022

economy considerations. As a starting point in this enterprise, we evaluate the predictions of the model for the processing preferences reported in recent attachment ambiguity studies for Italian (De Vincenzi and Job, 1993; Grillo and Costa, 2014).

2 Parsing Principles and RC Attachment Preferences

One of the most researched topics in the sentence processing literature is the cross-linguistic variation in *attachment ambiguity* preferences. Notoriously, when a complex Determiner Phrase (e.g., DP_1 *of* DP_2) is followed by a RC, languages are known to show varying biases for the RC modifying either DP_2 (Low Attachment, **LA**) or DP_1 (High Attachment, **HA**). Consider the following sentence:

(1) Pearl saw the Commander of the Gem that run

a. Pearl saw the Commander of [the Gem that run] **LA**

b. Pearl saw [[the Commander of the Gem] that run] **HA**

This sentence is ambiguous between two interpretations. In the LA interpretation, the relative clause [*that run*] modifies the second DP: e.g. in (1a), it is *the Gem* that is doing the running. However, a HA interpretation is also available (1b), according to which it is *the commander* that was running, with the RC modifying the whole complex DP [*the Commander of the Gem*]. While it is well established that English speakers will generally prefer the LA interpretation, it has been shown that languages vary significantly in this respect (Cuetos and Mitchell, 1988). For instance, Spanish, Greek, and Italian speakers show a general preference for a HA interpretation (Cuetos and Mitchell, 1988; Carreiras and Clifton Jr, 1993; De Vincenzi and Job, 1993; Papadopoulou and Clahsen, 2003, a.o.), while in Basque and Chinese speakers pattern similarly to English (Gutierrez-Ziardegi et al., 2004; Shen, 2006, a.o.). Additionally, variation in attachment preference within the same language has also been reported (Fernández, 2003), as well as variation across online and offline tasks (De Vincenzi and Job, 1993).

This cross-linguistic variation in Relative Clause (RC) attachment preferences has been the object of extensive investigation in theoretical linguistics and psycholinguistics. A long-standing hypothesis in the sentence processing literature has been that *processing economy* principles are a core feature of the human parser. In this sense, RC attachment ambiguity is of interest as a perfect case study for the exploration of such general economy considerations. As mentioned, such ambiguity is due to the possibility of attaching the RC to either the first DP or the second DP. An intuitive interpretation of locality of structure building would favor the latter, under the assumption that local attachment reduced the processing load of the parser (Frazier, 1990; Gibson et al., 1996; Gibson, 1998). While LA languages perfectly conform to the predictions made by such an hypothesis, HA languages present a problem. Importantly, numerous studies have unveiled a variety of factors that can modulate RC attachment — such as prosodic, semantic, and pragmatic variables (MacDonald et al., 1994; Gilboy et al., 1995; Acuna-Farina et al., 2009; Fernández, 2005; Fraga et al., 2005; Hemforth et al., 2015). These additional variables seem to behave somewhat consistently across languages, and the many existing proposals in the literature (Cuetos and Mitchell, 1988; Clifton Jr and Frazier, 1996; Gibson et al., 1996; Hemforth et al., 2000, a.o.) still leave the full pattern of cross-linguistic variation somewhat unexplained. If the goal is to provide explanatory insights into parsing mechanisms, even accounts that try to reduce variation in cross-linguistic preferences to statistical/exposure distributions would need to address whether HA/LA is less frequent in a specific language because of the inherent complexity of one construction over the other, or because of some external reason. All else being equal then, if the variance in the interpretative biases for RC attachment is not to be located in language specific grammatical distinctions, it might pose an issue for theories of language processing that see universal parsing mechanisms underlying human language processing behavior (see Grillo and Costa, 2014; Grillo et al., 2015; Aguilar et al., 2021, for a discussion).

While acknowledging the complicated array of variables affecting RC interpretation, Grillo and Costa (2014) point out a possible confounding factor in previous experiments reporting HA in languages like Italian and Spanish: namely, the availability of a *pseudo-relative* interpretation. Their claim is that in HA languages there is an additional structural representation available for sentences like (1): a *pseudo-relative* clause (PR) construction. Although linearly identical to RCs, PRs have different structural and semantic properties — essentially, they behave as NP/DP modifiers denoting events.

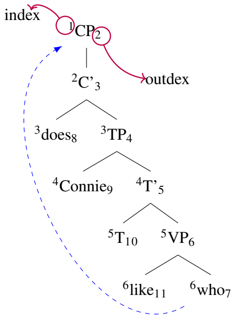
Importantly, the main structural difference is that in a PR parse, the matrix verb takes the whole PR as its complement (akin to what happens for English Small Clauses), and what looks like the "modified" DP is the subject of that clause. As PRs are complement/adjuncts of VPs, the most local DP is not grammatically available to the PR, and thus they are only compatible with what looks like a HA interpretation. Interestingly, it is possible to control for PR availability by modulating the syntactic and semantic environment of a sentence. With these considerations in mind, Grillo and Costa (2014) report that when participants are tested with sentences for which the RC interpretation is the only possible one (i.e., PR is made unavailable based on the properties of the main clause verb), a LA parse is preferred over the HA one (see also De Vincenzi and Job, 1993; Branco-Moreno, 2014; Aguilar et al., 2021).

These facts are accounted for by formulating a **pseudo-relative first** hypothesis. This hypothesis states that, when faced with a sentence ambiguous between a PR and a RC interpretation, the parser prefers committing to a PR structure first, thus giving rise to what looks like a HA preference. Then, if a PR analysis is made unavailable, the parser will prefer the LA parse over the HA due to universal locality principles. Grillo and Costa (2014) argue that the parsing preference for PR constructions might be due to the richer functional domain usually associated to RCs, making the latter dispreferred.

The preference of the parser for a PR structure is thus accounted for in this literature in terms of *simplicity* of the PR structure, and overall *economy* principles. However, while the locality ideas that would lead to preferring a LA over a HA have been extensively discussed in the past, the specific parsing principle grounding the alleged PR vs. RC complexity asymmetry is left generally unspecified. While the idea that structure building operations correspond to some type of cognitive cost is certainly not new, it is unclear that simply postulating additional functional structure *per se* implies increased parsing cost (Miller and Chomsky, 1963; Bresnan, 1978). In fact, it is possible to conceive of pletora of ways in which a specific structure could be defined as being simpler than another, and none of these are guaranteed to have concrete effects on a specific parsing strategy (Bresnan, 1982; Berwick and Weinberg, 1983). If such hypotheses are to be thoroughly explored, it seems crucial to ground our theoretical stipulations in a transparent theory of exactly why certain operations are more costly for the parser than others. In the rest of the paper, we propose the use of a computational model grounded in a rich grammar formalism, as a way to evaluate the economy claims made by the PR-first hypothesis. The following section illustrates the core ideas behind the model, and clarifies why such an approach can offer insights when testing theories of structural and processing complexity. For recent, detailed overviews of the technical details of the approach the reader is referred to (Gerth, 2015; Graf et al., 2017; De Santo, 2020b).

Steps	**Parse Action**
1	CP is conjectured
2	CP expands to C'
3	C's expands to does and TP
4	TP expands to Connie and T'
5	T' expands to T and VP
6	VP expands to like and who
7	**who** is found
8	**does** is found
9	**Connie** is found
10	**T** is found
11	**like** is found

(a)



(b)

Figure 1: Example of a string-driven top-down tree traversal for an MG derivation tree.

3 MG Parsing

MGs (Stabler, 1996, 2011) are a lexicalized formalism rigorously implementing an early version of Minimalist syntax. These grammars consist of a sets of lexical items (LIs), each with a phonetic form and a finite, non-empty string of features. Syntactic objects are built from LIs via two feature checking operations: *Merge* — encoding subcategorization — and *Move* — allowing for long-distance movement dependencies. In this paper, we will ignore the feature component of the LIs, and focus on the fact that the fundamental data structure in MGs is a *derivation tree*, which encodes the sequence of Merge and Move operations required to build the phrase structure tree for a given sentence (Michaelis, 1998; Harkema, 2001; Kobele et al., 2007).

Merge and Move operations are represented in these trees as binary and unary branching nodes, respectively. The main difference between a more traditional phrase structure tree and a derivation tree is that in the latter, the final word order of a sentence is not directly reflected in the order of the leaf nodes in a derivation tree. This is because moving phrases remain in their base position, and their landing site can be deterministically reconstructed via the feature calculus.

Given that MGs are able to represent the structurally rich analyses now common in Minimalist syntax, they have been focus of a line of work aimed at connecting syntactic assumptions to offline processing behavior. Specifically, this work has shown that a top-down parser for MGs (Stabler, 2013) can successfully predict a variety of processing difficulty contrasts, via metrics that relate offline parsing difficulty to memory usage (Kobele et al., 2013; Graf et al., 2017; De Santo, 2020b, a.o.).

Stabler (2013)'s parser is a variant of a standard recursive-descent parser for CFG, modified to take care of the fact that the order of lexical items in a derivation tree does not fully match the linear surface order. Basically, the parser scans the nodes from top to bottom and from left to right; but since the surface order of lexical items in the derivation tree is not the phrase structure tree's surface order, simple left-to-right scanning of the leaf nodes yields the wrong word order. In order to keep track of the derivational operations affecting the linear word order, the MG variant follows the standard approach of predicting nodes downward (toward words) and left-to-right until a Move node is predicted. At that point, the parser discards the top-down strategy and builds the shortest path towards the predicted mover. After the mover has been found, the parser continue from the point where the search for the mover started (Figure 1a). The memory stack associated to the parser therefore plays a fundamental role: if a node is hypothesized at step i, but cannot be worked on until step j, it must be stored for $j-i$ steps in a priority queue. For instance, considering the derivation tree in Figure 1b, the node for *does* is predicted at step 3. However, since a movement dependency for Spec,CP has been postulated, the parser is not following a pure top-down strategy and will not match that prediction against the linear input until a node for *who* has been predicted and confirmed (at step 6 and 7).

To make the traversal strategy easy to follow, we adopt Kobele et al. (2013)'s tree annotation approach. The annotation indicates for each node in the tree when it is first conjectured by the parser (*index*, superscript) and placed in the memory queue, and at what point it is considered completed and flushed from memory (*outdex*, subscript). Index and Outdex allow the *MG model* to rigorously link parser behavior, syntactic structure, and processing difficulty by connecting the stack states of the top-down parser to memory usage. In order to allow for psycholinguistic predictions, it is then possible to use these annotations to predict processing difficulty based on how the structure of a derivation tree affects memory usage during a parse (Rambow and Joshi, 1994; Gibson, 2000; Kobele et al., 2013; Graf and Marcinek, 2014; Gerth, 2015).

The MG model distinguishes several cognitive notions of memory usage (Graf et al., 2017). Here, we focus on a measure of how long a node is kept in memory through a derivation (TENURE). Tenure for each node is computed considering the moment a node was first postulated into the structure (and thus placed in the memory stack of the parser) and the moment such prediction was confirmed and the node could be taken out of memory. Essentially then, a node's tenure is equal to the difference between its index and its outdex. For instance, considering the annotated MG tree in Figure 1b, tenure for *Connie* is $Outdex(Connie) - Index(Connie) = 9 - 4 = 5$.

Based on how this cognitive notion of memory usage interacts with the geometry of the underlying syntactic structure, the MG parser then assigns a cost to each sentence. Kobele et al. (2013) show that tenure can be associated to quantitative values by defining metrics like MAXT $:= max(\{tenure\text{-}of(n)\})$ and SUMT $:= \sum_n tenure\text{-}of(n)$. MAXT measures the maximum amount of time any node stays in memory during processing, while SUMT measures the overall amount of memory usage for all nodes whose tenure is not trivial (i.e., >2). It thus captures total memory usage over the course of a parse. A metric like MAXT can then be used to derive categorical processing contrasts, by comparing the tenure values assigned by the MG model to derivation trees corresponding to sentences with stark asymmetries in reported offline processing preferences. For instance, building on these intuitions, Graf and Marcinek (2014) show that MAXT makes the right difficulty predictions for several phenomena, such as right embedding vs. center embedding, nested dependencies vs. crossing dependencies, as well as a set of cross-linguitic contrasts involving relative clauses. Importantly, while the space of possible metrics defined by this model is potentially vast, in what follows we will focus our discussion on MAXT exclusively, given the attention that this specific metric has received in recent work (Gerth, 2015; Graf et al., 2017; Liu, 2018; Lee, 2018; De Santo, 2019, 2020a).

Finally, note that Stabler's original parser is equipped with a search beam discarding the most unlikely predictions. Consistent with previous work, we follow Kobele et al. (2013) in ignoring the beam and assuming that the parser is equipped with a perfect oracle, which always makes the right choices when constructing a tree. This idealization is clearly implausible from a psycholinguistic point of view, and might seem controversial when modeling structurally ambiguous sentences. However, it is made with a precise purpose in mind: by assuming a deterministic parse, we aim to evaluate structural economy claims by focusing on the specific contribution of syntactic complexity to memory load.

4 PRs vs RCs in a Computational Model

Consider now the following Italian sentence:

(2) (Io) Ho visto la nonna della ragazza
(I) have seen the grandma of the girl
che gridava
that screaming-3SG

"I saw the grandma of the girl that was screaming"

This sentence is ambiguous between a HA interpretation (*the grandma was screaming*) and a LA interpretation (*the girl was screaming*). Additionally, the HA interpretation is ambiguous between two structural analyses: a PR analysis and a true HA, RC analysis.

As mentioned, the *pseudo-relative first* hypothesis as stated in previous literature predicts that a PR parse should be preferred over RC parses (both LA and HA), due to the overall simplicity of PRs over RCs. Additionally, the hypothesis then predicts that, in absence of an available PR parse, a LA parse should be preferred over an HA one, possibly due to locality principles. The relevant pairwise contrasts are summarized in Table 1. Note that, while the PR $<$ LA[1] contrast might seem counter intuitive, it is crucial for the PR-first hypothesis to pan out: when faced with a choice, the parser follows a PR strategy first as it is (in some ways) simpler. A conceivable weaker version of the hypothesis, which makes a prediction only for the HA vs. PR contrast with nothing to say about the relative simplicity of the PR structure when compared to relative clauses with LA constructions would be insufficient, as it would not explain why the parser does not follow a LA strategy to begin with. With this in mind, we test this hypothesis over sentences in Italian as reported in (De Vincenzi and Job, 1993; Grillo and Costa, 2014).

4.1 Syntactic Choices

Adopting MGs as the core grammar formalism makes the model sensitive to fine-grained syntactic choices. Exploring how different syntactic analyses impact the main results is thus important to the explanatory aims of the approach (De Santo, 2021). In this sense, the psycholinguistic literature tends to be fairly non-committal with respect to the details of the structural hypotheses underlying relative clauses. Therefore, here we evaluate two different analyses of RC constructions currently popular in minimalist syntax (Bianchi, 2002a,b): the promotion analysis (Kayne, 1994) and the *wh*-movement analysis (Chomsky, 1977).

Promotion Analysis Under a promotion analysis (Kayne, 1994), the head of the RC is a noun starting out as an argument of the embedded verb and undergoing movement into the specifier of the RC. The RC itself is selected by the determiner that would normally select the head noun in head-external accounts, like the *wh*-movement case below (Figure 2a).

***Wh*-movement Analysis** Chomsky (1977)'s *wh*-movement analysis treats the construction of an RC as an instance of *wh*-movement. The complementizer position is overtly filled by *that*, while a silent wh-operator *Op* moves from the base position to Spec,CP. The whole CP merges with the relativized NP as its adjunct (Figure 2b). The silent *Op* is co-indexed with the NP to which the RC is adjoining.

Pseudo-relatives Similarly to RCs, there are various potential analyses to pseudo-relative clause constructions. Here, we follow Grillo and Costa (2014) and adopt an approach to PR structures as small clauses (Cinque, 1992). Essentially, as mentioned before, in PR parses the matrix verb takes the whole PR as its complement, and the modified DP is the head of that clause (Figure 2c). Thus, there is no movement extracting the head DP from within the PR. The modified DP is linked to its interpreted position by co-indexing it with a null *pro*, resembling what is done with RC in the *wh*-movement analysis.

[1] Henceforth in the paper, processing contrasts are summarized as x $<$ y, to be interpreted as x is preferred over y.

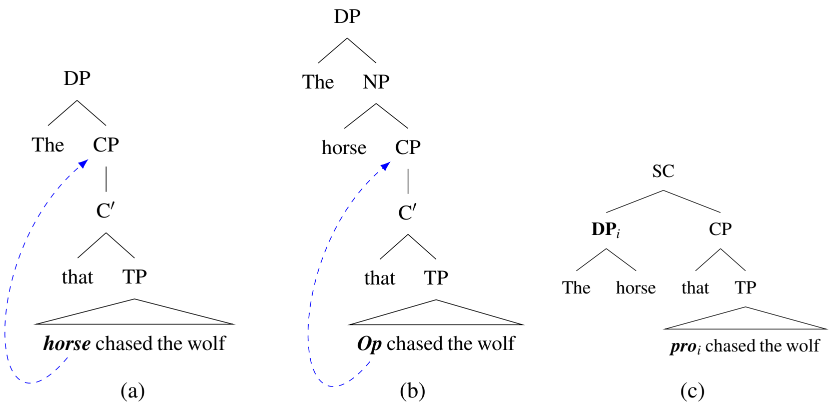


Figure 2: Sketches of the *(a)* RC with promotion, *(b)* RC with *wh*-movement, and *(c)* PR analyses for the sentence *The horse that the wolf chased.*

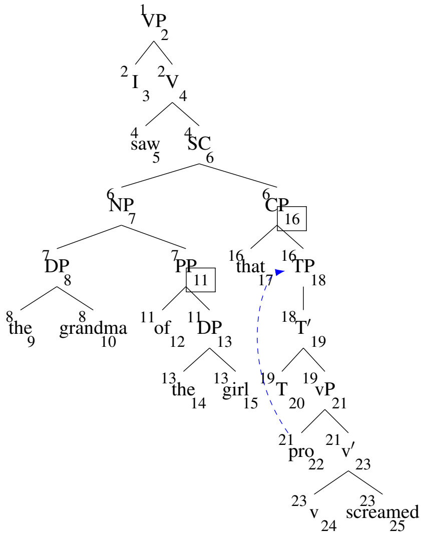


Figure 3: Annotated derivation trees for the Italian sentence *I saw the grandma of the girl that screamed*, according to a pseudo-relative clause analysis. The root of the tree is treated as a VP since additional structure in the matrix clause would be identical across comparisons. Boxed nodes are those with tenure value greater than 2, following (Graf and Marcinek, 2014).

4.2 Modeling Results[2]

As the model's results are categorical, only one test item per construction is needed. Consider once again the ambiguous sentence in (4). According to what discussed in the previous section, we can build five derivations for that single linear string: one derivation using a PR analysis, and then two derivations for RC/LA and RC/HA, each modulated by syntactic analysis (*wh*-movement or promotion). Annotated derivation trees for these configurations can be seen in Figure 3 and Figure 4. With all of this in place, we can finally look at the modeling results. Table 1 and Table 2 report overall performance of the model, and MAXT values for the three constructions considered here.

First, MAXT successfully captures the $LA < HA$ preference, independently of syntactic analysis. This is because in the HA cases the parser has to search for the whole NP [*the grandma of the girl*] before being able to work on the rest of the RC (Figures 4a and 4b vs. Figures 4c and 4d). This is encouraging, in the sense that it shows how the model captures the well-established intuition about locality of attachment for these two constructions.

We can then move on to the pseudo-relative contrasts. Under a promotion analysis, the parser correctly captures $PR < HA$, due to the additional movement dependencies hypothesized for the RC/HA structure (Figure 4a). However, MAXT predicts no difference between the two when the RC is built according to a *wh*-movement approach (Figure 4b). Looking at the annotated derivation trees for the HA case, it is possible to infer that in the promotion case MAXT (measured on the complementizer *that*) is driven by the fact that the whole head NP raises to Spec,CP. Thus, the parser needs to expand it in its base position (Spec,*v*P) before being able to work on the rest of the CP. This contrasts starkly with what is done when building the PR structure: since there is no movement dependency to resolve, having to build the big NP first does weight on the CP node somewhat, but it does not affect how long CP internal nodes have to be maintained in memory

[2] All simulations in this paper were run on the open source code made available by Graf et al. (2017) at https://github.com/CompLab-StonyBrook/mgproc.

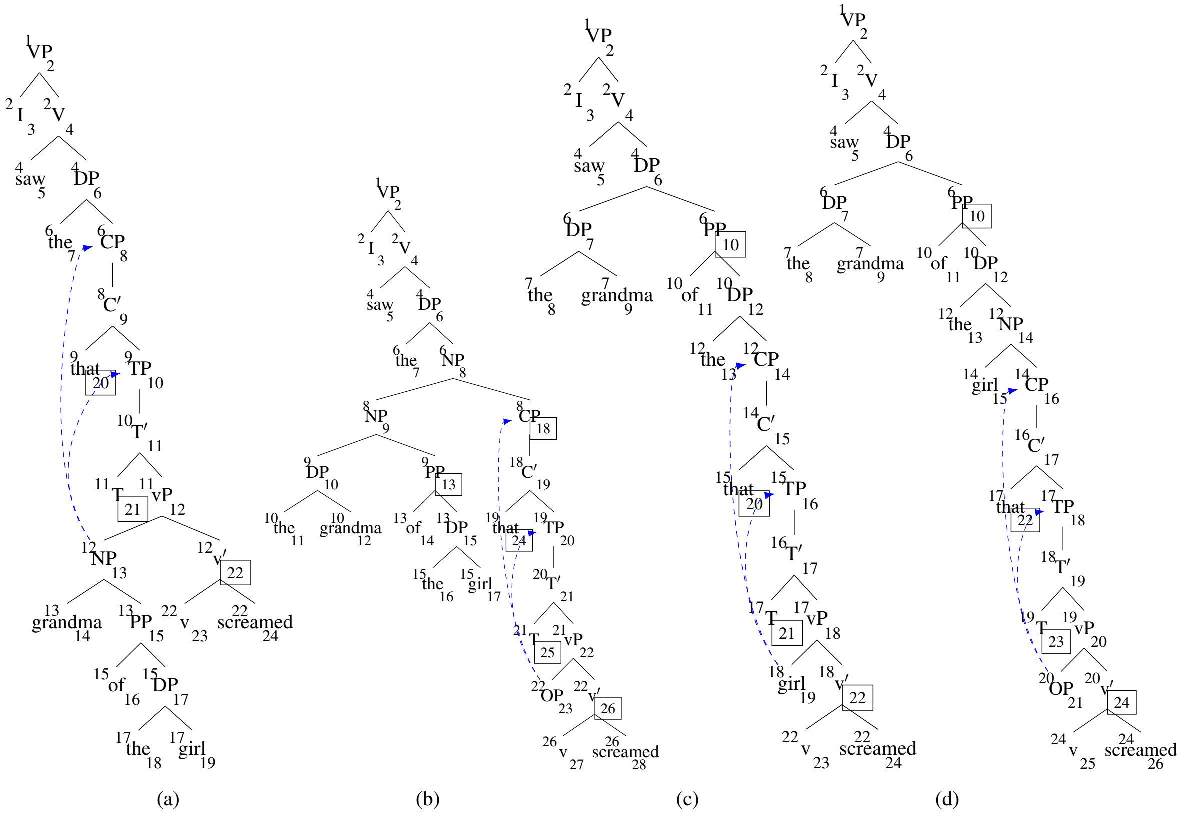


Figure 4: Annotated derivation trees for Italian Relative Clauses: *(a)* HA with a promotion analysis, *(b)* HA attachment with a *wh*-movement analysis, *(c)* LA with a promotion analysis, and *(d)* LA with a *wh*-movement analysis for the sentence *I saw the grandma of the girl that screamed.* Trees are treated as VPs since additional structure in the matrix clause would be identical across comparisons. Boxed nodes are those with tenure value greater than 2, following (Graf and Marcinek, 2014).

(Figure 3). Crucially though, this is very similar to what has to be done for RCs according to the *wh*-movement analysis. According to this approach, there is no movement of the whole NP from within the RC, but just of an operator to Spec,CP. Thus, while there are some subtle structural differences between RCs and PRs under the *wh*-movement analysis too, they do not end up affecting overall memory load in any significant way (beyond the specific node on which MAXT is measured).

Finally, we look at the last contrast relevant to the PR-first hypothesis. Under neither of the RC analyses considered the model is able to capture the fact that a PR construction should be more efficient to parse than a LA attachment RC one. This is because for both PR and HA structures, the parser has to explore the full complex NP before being able to expand on the rest of the structure (thus increasing memory load on the hypothesized embedded CP), while in the LA case only the lower DP needs to be fully built and discarded from memory.

	MG Parser	
Hypothesis	**Promotion**	**Wh-mov**
PR < HA	✓	Tie
PR < LA	×	×
LA < HA	✓	✓

Table 1: Summary of the predictions made by a *pseudo-relative first* account, and corresponding parser's predictions based on MAXT, as pairwise comparisons (x < y: x is preferred over y).

5 Conclusion

In this paper, we exploited a transparent computational model connecting grammatical representations to memory cost via parsing, in order to explicitly test an economy-based hypothesis about why pseudo-relative clauses are preferred over relative clauses in psycholinguistic experiments. This PR-first hypothesis has been put forward in the literature as a way to account for the reported cross-linguistic variation between high-attachment

	MAXT	
	Promotion	**Wh-mov**
PR	10/CP	
HA	11/that	10/CP
LA	5/that	7/that

Table 2: MAXT values (*value/node*) by construction, with RCs modulated across a promotion and *wh*-movement analysis.

(HA) and low-attachment (LA) parsing preferences — which would then arise as an artifact of a syntactic difference between languages with pseudo-relative constructions and languages without it. In order to evaluate the broader implications of the PR-first idea then, what seems crucial is the ability to explore the soundness of the complexity predictions made by this hypothesis when interacting with the broad range of fine-grained syntactic assumptions for a minimalist derivation.

Using complexity metrics calculated on Minimalist Grammar derivations for Italian sentences, we showed that a preference for PR over HA is predicted, as well as the more traditional preference for LA over HA, but the required preference for PR over LA is not. Overall then, our results do not support a memory-based, parsing economy explanation for a PR preference in Italian.

Importantly, these modeling results do not call into question the strong experimental evidence for PR availability modulating attachment preferences (De Vincenzi and Job, 1995; Branco-Moreno, 2014; Grillo and Costa, 2014; Grillo et al., 2015; Aguilar et al., 2021, a.o.), and thus do not weaken the analysis of LA/HA variation as an artifact of PR availability per se. In fact, the MG model's predictions are *consistent* with the idea that, when comparing genuine RC structures, a LA derivation should be easier than a HA derivation. What these results invite us to consider however, is the importance of deeper evaluations of "simplistic" explanations of processing facts based on un-specified parsing simplicity principles. While our model might not tell us *why* PRs are preferred over RCs, it suggests ways to narrow down the space of plausible accounts.

Obviously, there are a variety of ways in which simplicity claims can be incorporated into a parsing model (Boston, 2012). Moreover, here we only considered structural differences between PRs and RCs while, as Grillo and Costa (2014) themselves suggest , notions of complexity driven by semantic/pragmatic differences might be playing an important role (Crain, 1985; Altmann and Steedman, 1988).

Cross-linguistic validation is also fundamental in this type of inquiry. Note that under standard assumptions a corresponding Spanish sentence would only differ lexically for the Italian cases, and the numerical contrasts would be virtually identical across the two languages. Thus, while the discussion in this paper was focused on Italian, these results straightforwardly extend to Spanish too. However, in the future it will be important to extend this evaluation to HA languages with wider syntactic differences, for which a PR advantage has been also established experimentally (e.g., French; Koenig and Lambrecht, 1999; Pozniak et al., 2019). In this sense, the pairwise differences needed by the MG parser might also suggest ways to design fine-grained experimental contrasts for languages in which PR availability still lacks experimental support.

Finally, the difference between the performance under a promotion vs. *wh*-movement account highlights once again the model's sensitivity to syntactic details, and reveals how different syntactic choices might affect notions of *simplicity* grounded in parsing intuitions in unexpected ways. Crucially though, what our results reveal is that quantified implementations of simplicity and economy might differ significantly from more broadly specified, general intuitions. Transparent computational models, coupled with more extensive cross-linguistic experimental comparisons, can then play a crucial role in building theories of the interface between grammatical principles and sentence processing mechanisms that are explicit and explanatory.

Acknowledgments

The authors would like to thank SCiL 2022 anonymous reviewers for their helpful suggestions.

References

Carlos Acuna-Farina, Isabel Fraga, Javier García-Orza, and Ana Piñeiro. 2009. Animacy in the adjunction of spanish rcs to complex nps. *European Journal of Cognitive Psychology*, 21(8):1137–1165.

Miriam Aguilar, Pilar Ferré, José M Gavilán, José A Hinojosa, and Josep Demestre. 2021. The actress was on the balcony, after all: Eye-tracking locality and pr-availability effects in spanish. *Cognition*, 211:104624.

Gerry Altmann and Mark Steedman. 1988. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Robert C. Berwick and Amy S. Weinberg. 1983. The role of grammar in models of language use. *Cognition*, 13:1–61.

Valentina Bianchi. 2002a. Headed relative clauses in generative syntax. part i. *Glot International*, 6(7):197–204.

Valentina Bianchi. 2002b. Headed relative clauses in generative syntax. part ii. *Glot International*, 6(8):1–12.

Željko Bošković and Troy Messick. 2017. Derivational economy in syntax and semantics. In *Oxford research encyclopedia of linguistics*.

Marisa Ferrara Boston. 2012. *A computational model of cognitive constraints in syntactic locality*. Ph.D. thesis, Cornell University.

David Branco-Moreno. 2014. *The influence of pseudo-relatives on attachment preferences in Spanish*. City University of New York.

J. Bresnan. 1982. *The Mental representation of grammatical relations*. MIT Press series on cognitive theory and mental representation. MIT Press.

Joan Bresnan. 1978. A realistic transformational grammar. *Linguistic theory and psychological reality*, pages 1–59.

Manuel Carreiras and Charles Clifton Jr. 1993. Relative clause interpretation preferences in spanish and english. *Language and speech*, 36(4):353–372.

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

Noam Chomsky. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Guglielmo Cinque. 1992. The pseudo-relative and acc-ing constructions after verbs of perception. *Working Papers in Linguistics,¡ 2¿, 1992, pp. 1-31*.

Charles Clifton Jr and Lyn Frazier. 1996. Construal.

Chris Collins. 2001. *Economy conditions in syntax*. na.

Stephen Crain. 1985. On not being led up the garden path. *Natural language parsing*, pages 320–358.

Fernando Cuetos and Don C Mitchell. 1988. Cross-linguistic differences in parsing: Restrictions on the use of the late closure strategy in spanish. *Cognition*, 30(1):73–105.

Aniello De Santo. 2019. Testing a Minimalist grammar parser on Italian relative clause asymmetries. In *Proceedings of the ACL Workshop on Cognitive Modeling and Computational Linguistics (CMCL) 2019*, June 6 2019, Minneapolis, Minnesota.

Aniello De Santo. 2020a. Mg parsing as a model of gradient acceptability in syntactic islands. *Proceedings of the Society for Computation in Linguistics*, 3(1):53–63.

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

Aniello De Santo. 2021. Italian postverbal subjects from a minimalist parsing perspective. *Lingue e linguaggio*, 20(2):199–227.

Marcia De Vincenzi and Remo Job. 1995. An investigation of late closure: The role of syntax, thematic structure, and pragmatics in initial interpretation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21(5):1303.

Marica De Vincenzi. 1991. *Syntactic parsing strategies in Italian: The minimal chain principle*, volume 12. Springer Science & Business Media.

Marica De Vincenzi and Remo Job. 1993. Some observations on the universality of the late-closure strategy. *Journal of Psycholinguistic Research*, 22(2):189–206.

Vera Demberg and Frank Keller. 2009. A computational model of prediction in human parsing: Unifying locality and surprisal effects. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 31.

E Fernández. 2005. The prosody produced by spanish-english bilinguals: A preliminary investigation and implications for sentence processing. *Revista da ABRALIN*, 4(1):109–141.

Eva Fernández. 2003. Bilingual sentence processing. *Amsterdam: John Benjamins. Ferreira, F.(2003). The misinterpretation of noncanonical sentences. Cognitive Psychology*, 47:164–203.

Isabel Fraga, Javier García-Orza, and Juan Carlos Acuña. 2005. La desambiguación de oraciones de relativo en gallego: Nueva evidencia de adjunción alta en lenguas romances. *Psicológica*, 26(2):243–260.

Lyn Frazier. 1978. On comprehending sentences: Syntactic parsing strategies. *Doctoral dissertation, University of Connecticut*.

Lyn Frazier. 1987. Syntactic processing: evidence from Dutch. *Natural Language & Linguistic Theory*, 5(4):519–559.

Lyn Frazier. 1990. Parsing modifiers: Special purpose routines in the human sentence processing mechanism. *Comprehension processes in reading*, pages 303–330.

Minoru Fukuda. 2011. Locality in minimalist syntax by thomas s. stroik, linguistic inquiry monographs 51, mit press, cambridge, ma, 2009, x+ 149pp. *ENGLISH LINGUISTICS*, 28(2):321–333.

Sabrina Gerth. 2015. *Memory Limitations in Sentence Comprehension: A Structural-based Complexity Metric of Processing Difficulty*, volume 6. Universitätsverlag Potsdam.

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. In *2000, Image, Language, Brain: Papers from the First Mind Articulation Project Symposium*, pages 95–126. MIT press.

Edward Gibson, Neal Pearlmutter, Enriqueta Canseco-Gonzalez, and Gregory Hickok. 1996. Recency preference in the human sentence processing mechanism. *Cognition*, 59(1):23–59.

Elizabeth Gilboy, Josep-MMaria Sopena, Charles Cliftrn Jr, and Lyn Frazier. 1995. Argument structure and association preferences in spanish and english complex nps. *Cognition*, 54(2):131–167.

Thomas Graf and Bradley Marcinek. 2014. Evaluating evaluation metrics for minimalist parsing. In *Proceedings of the 2014 ACL Workshop on Cognitive Modeling and Computational Linguistics*, pages 28–36.

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

Nino Grillo and João Costa. 2014. A novel argument for the universality of parsing principles. *Cognition*, 133(1):156–187.

Nino Grillo, João Costa, Bruno Fernandes, and Andrea Santi. 2015. Highs and lows in english attachment. *Cognition*, 144:116–122.

E Gutierrez-Ziardegi, Manuel Carreiras, and Itziar Laka. 2004. Bilingual sentence processing: relative clause attachment in basque and spanish. In *ANNUAL CUNY CONFERENCE ON HUMAN SENTENCE PROCESSING*, volume 17.

Henk Harkema. 2001. A characterization of minimalist languages. In *International Conference on Logical Aspects of Computational Linguistics*, pages 193–211. Springer.

Barbara Hemforth, Susana Fernandez, Charles Clifton Jr, Lyn Frazier, Lars Konieczny, and Michael Walter. 2015. Relative clause attachment in german, english, spanish and french: Effects of position and length. *Lingua*, 166:43–64.

Barbara Hemforth, Lars Konieczny, and Christoph Scheepers. 2000. Syntactic attachment and anaphor resolution: The two sides of relative clause attachment.

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

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

Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In *Model Theoretic Syntax at 10*, pages 73–82. J. Rogers and S. Kepser.

Jean-Pierre Koenig and Knud Lambrecht. 1999. French relative clauses as secondary predicates: A case study in construction theory. In *Unpublished ms. Communication préeseentée au 2e Colloque de Syntaxe et Smantique de Paris*, volume 17.

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

Lei Liu. 2018. Minimalist Parsing of Heavy NP Shift. In *Proceedings of PACLIC 32 The 32nd Pacific Asia Conference on Language, Information and Computation*, The Hong Kong Polytechnic University, Hong Kong SAR.

Maryellen C MacDonald, Neal J Pearlmutter, and Mark S Seidenberg. 1994. The lexical nature of syntactic ambiguity resolution. *Psychological review*, 101(4):676.

Jens Michaelis. 1998. Derivational minimalism is mildly context–sensitive. In *International Conference on Logical Aspects of Computational Linguistics*, pages 179–198. Springer.

George A. Miller and Noam Chomsky. 1963. Finitary models of language users. In R. Luce, R. Bush, and E. Galanter, editors, *Handbook of Mathematical Psychology*, volume 2. John Wiley, New York.

Alexandra Motut. 2010. Merge over move and the empirical force of economy in minimalism. *Toronto Working Papers in Linguistics*, 33.

Despina Papadopoulou and Harald Clahsen. 2003. Parsing strategies in l1 and l2 sentence processing: A study of relative clause attachment in greek. *Studies in Second Language Acquisition*, 25(4):501–528.

Céline Pozniak, Barbara Hemforth, Yair Haendler, Andrea Santi, and Nino Grillo. 2019. Seeing events vs. entities: The processing advantage of pseudo relatives over relative clauses. *Journal of Memory and Language*, 107:128–151.

Owen Rambow and Aravind K Joshi. 1994. A processing model for free word order languages. *Perspectives on Sentence Processing*.

Maryam Razaghi, Shahin Rahavard, and Firooz Sadighi. 2015. Economy, simplicity and uniformity in minimalist syntax. *International Journal on Studies in English Language and Literature*.

Xingjia Shen. 2006. Late assignment of syntax theory: evidence from chinese and english.

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

Edward P Stabler. 2011. Computational perspectives on minimalism. In *The Oxford Handbook of Linguistic Minimalism*.

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

How well do LSTM language models learn filler-gap dependencies?

Satoru Ozaki, Dan Yurovsky, Lori Levin
Carnegie Mellon University
{ ikazos, yurovsky }@cmu.edu, levin@andrew.cmu.edu

Abstract

This paper revisits the question of what LSTMs know about the syntax of filler-gap dependencies in English. One contribution of this paper is to adjust the metrics used by Wilcox et al. (2018) and show that their language models (LMs) learn embedded *wh*-questions – a kind of filler-gap dependencies – better than they originally claimed. Another contribution of this paper is to examine four additional filler-gap dependency constructions to see whether LMs perform equally on all types of filler-gap dependencies. We find that different constructions are learned to different extents, and there is a correlation between performance and frequency of constructions in the Penn Treebank Wall Street Journal corpus.

1 Introduction

Language models (LMs) that use recurrent neural networks (RNNs, Elman, 1990), especially those adopting the long short-term memory (LSTM, Hochreiter and Schmidhuber, 1997) architecture, achieve outstanding performance in various natural language processing tasks. The fact that the same architecture yields high performance across many tasks seems to suggest that these LMs are learning something fundamental about natural language.

But what does it mean to learn a language, and have neural networks really achieved language acquisition? Much recent work focuses on evaluating neural networks' understanding of various syntactic phenomena that occur in natural language, such as subject-verb agreement (Linzen et al., 2016; Bernardy and Lappin, 2017; Kuncoro et al., 2018; Gulordava et al., 2018), negative polarity item licensing (Futrell et al., 2018; Jumelet and Hupkes, 2018; Marvin and Linzen, 2018), and anaphora (Marvin and Linzen, 2018; Warstadt et al., 2019).[1] These studies typically take a pre-trained LM or train one from scratch, and test the LM's performance on a dataset of artificially constructed linguistic expressions or a curated subset of real-world linguistic utterances, which pertain to particular syntactic phenomena of the researcher's interest.

Following Chowdhury and Zamparelli (2018), Wilcox et al. (2018, 2019b) and others, we focus on English filler-gap dependencies because of their three interesting properties: (a) bijectivity of filler and gap, (b) unboundedness, and (c) sensitivity to island constraints. We will review these in more detail in Section 2.

Wilcox et al. (2018) address whether neural networks know the bijectivity property: fillers are bad without gaps and gaps are bad without fillers. Their LMs detect that a filler is better (less surprising) with a gap than without a gap, but they do not fully capture bijectivity. One contribution of our paper is to experiment with different types of probabilistic metrics. With our changes, we show that Wilcox et al.'s models do in fact fully capture the bijectivity property for one metric.

English has a variety of filler-gap dependency constructions, which share the same three properties. While some linguists have analyzed these constructions as generated from a common abstract syntactic mechanism such as *wh*-movement (Chomsky, 1977), others have analyzed them as a mixture of idiosyncractic constructions (Sag, 2010). Do LMs capture the same properties across all constructions, or does performance vary over constructions? In this paper, we extend the work of Wilcox et al. (2018) to include four additional filler-gap dependency constructions, and examine their behavior collectively and individually to see how they bear on the issues of general mechanisms and specific constructions in human language.

[1] For a summary of the types of syntactic phenomena tested and the list of studies in the literature for each type, see Warstadt et al. (2020).

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 76-88.
Held on-line February 7-9, 2022

This paper is structured as follows. In Section 2, we review the properties of English filler-gap dependencies and previous work on RNNs' acquisition of filler-gap dependencies. In Section 3, we revisit Wilcox et al. (2018)'s experiment, revise their metrics and propose stricter criteria for the acquisition of filler-gap dependencies. We show that for one metric, their LMs understand filler-gap dependencies better than they had previously claimed. In Section 4, we check if LMs learn four other kinds of filler-gap dependency constructions, and their interaction with island constraints and embedding depth. We see that different constructions are learned to different extents. In Section 5, we test if the performance for each type of construction we obtain in Section 4 correlates with the relative frequency of these constructions in a written-text English corpus. Finally, in Section 6, we conclude our findings.

2 Properties of English Filler-Gap Dependencies

(1-a) is an example of a *wh*-question, which is a filler-gap dependency construction in English. The verb *put* is followed by a **gap**, indicated by an underscore, which is an empty position that would canonically be occupied as in *We put the book on the table*. The word *what* is understood to fill the gap and is called the **filler**. The filler and the gap are marked by a common subscript index.

(1) a. **What**$_i$ did you put ___$_i$ on the table?
b. ***What**$_i$ did you put **it**$_i$ on the table?
c. *You put ___$_i$ on the table.
d. You put **it** on the table.

English has several kinds of filler-gap dependency constructions, including comparatives (2-a), *it*-clefts (2-b), topicalization (2-c), embedded *wh*-questions (2-d), *tough*-movement (2-e) and a few others (Chomsky, 1977; Huddleston and Pullum, 2002; Sag, 2010; Chaves and Putnam, 2021, among others).

(2) a. Maryanne read **more books**$_i$ this month than Alfred read ___$_i$ last month.
b. It was **Anna**$_i$ that Kevin talked to ___$_i$.
c. **These movies**$_i$, Antonio wishes he had never seen ___$_i$.
d. Someone figured out **who**$_i$ Margaret was describing ___$_i$.
e. **Thomas**$_i$ was difficult to persuade ___$_i$.

Filler-gap dependencies are of interest for at least three reasons. First is the property of bijectivity of filler and gap: there can be no gap without a filler and no filler without a gap. (1-b) is ungrammatical because there is a filler (*what*) but where we would expect a gap, there is a pronoun (*it*). Conversely, (1-c) is ungrammatical because there is a gap, but no filler.

Second, filler-gap constructions are unbounded in the sense that the filler and gap can be separated by a potentially unlimited number of clausal boundaries (three in (3)). This poses a challenge to language modelling, as these dependencies must be modelled robustly across arbitrarily many intervening words.

(3) **What**$_i$ did Rebecca believe [you and Albert said [the professor thought [she already discussed ___$_i$ last week]]] ?

Finally, the availability of filler-gap dependencies is constrained by complex structural restrictions. This is illustrated in (4-a), which is a paraphrase of (4-b). Though the two questions differ only minimally in their structure, (4-a) is ungrammatical while (4-b) is not. On the other hand, (4-c), which has the same structure as (4-a) but no filler-gap dependency, is grammatical. This shows that there is a constraint that disallows filler-gap dependencies across a kind of structure unique to (4-a). The precise identification and characterization of such constraints are challenging for linguists, and the mere existence of such constraints poses a challenge for language acquisition researchers: how do children acquire such complex structural linguistic rules from exposure to positive evidence alone?

(4) a. ***What**$_i$ did Rebecca believe your claim that the professor discussed ___$_i$?
b. **What**$_i$ did Rebecca believe you claimed that the professor discussed ___$_i$?
c. Did Rebecca believe your claim that the professor discussed **this**?

There is much debate on the question of how well RNNs learn filler-gap dependencies. Chowdhury and Zamparelli (2018) claim that GRU and LSTMs produce higher perplexity and cross-entropy loss for ungrammatical, gapless *wh*-questions than for their grammatical, gapped counterparts (e.g. *Which candidate should the students discuss ___/*him?*). However, their performances are heavily affected by sentence processing factors. Wilcox et al. (2018, 2019b, *et seq.*) look at two pre-trained LSTM LMs and define a metric called *wh*-licensing interaction, which measures the extent to which the surprisal of a gapped clause is reduced significantly by the presence of the licensor. Using this metric, they show their LMs learn several structural properties of filler-gap dependencies as well as certain island constraints. On the other hand, Da Costa and Chaves (2020) and Chaves (2020) study the same LMs with respect to number agreement between the head noun and the verb of a relative clause (e.g. *which lawyer I think was/*were ...*) and observe that the LMs become less sensitive to agreement violations as the dependency crosses increasing levels of embeddings. They also claim that the island constraints Wilcox et al. (2018) purport these LMs to learn have certain exceptions, which are not acquired by these LMs.

3 Study 1: Surprisal and grammaticality

Wilcox et al. (2018) use a 2x2 factorial design as in (5), differing by [licensor], i.e. the presence/absence of the licensor, and [gap], i.e. the presence/absence of the gap. Their data consists entirely of embedded *wh*-questions. The filler is called a **licensor** because the gap cannot occur without it.

(5) a. I know that the lion devoured a gazelle at sunrise. [-licensor, -gap]
b. *I know what the lion devoured a gazelle at sunrise. [+licensor, -gap]
c. *I know that the lion devoured ___ at sunrise. [-licensor, +gap]
d. I know what the lion devoured ___ at sunrise. [+licensor, +gap]

They experiment on two pre-trained LSTM LMs. The first is the **Google model** (Jozefowicz et al., 2016). Trained on the One Billion Word Benchmark (Chelba et al., 2013), it consists of two hidden layers with 8196 units each. The second is the **Gulordava model** (Gulordava et al., 2018). Trained on 90 million tokens of English Wikipedia, it consists of two hidden layers with 650 units each.

The metric designed by Wilcox et al. builds on the definition of surprisal in (6) (Hale, 2001; Levy, 2008; Smith and Levy, 2013), where $S(w_k)$ is the surprisal generated by an RNN upon seeing the word w_k in a sentence, and h_{k-1} is the RNN's hidden state after consuming all previous words in the sentence. The probability is calculated from the RNN's softmax activation.

(6) $S(w_k) = -\log_2 \mathbb{P}(w_k|h_{k-1})$

For each experimental item, Wilcox et al. measures surprisal at two places: summed over a region immediately following the potential gap (emphasized in (7-a)), and summed over the entire embedded clause following the potential licensor (emphasized in (7-b)). The former, which we call **local surprisal**, reflects any local effects from the gap's licitness, while the latter, which we call **global surprisal**, reflects global expectations about the general well-formedness of the sentence.

(7) a. I know that/what the lion devoured (a gazelle) *at sunrise* .
b. I know that/what *the lion devoured (a gazelle) at sunrise* .

One can thus extend the definition of surprisal to be a function of experimental items, i.e. sentences. Then, Wilcox et al. define a metric they call *wh*-licensing interaction, as $(S(\text{[+licensor, -gap]}) - S(\text{[-licensor, -gap]})) - (S(\text{[+licensor, +gap]}) - S(\text{[-licensor, +gap]}))$.

This metric computes the surprisal difference between the two kinds of sentences that are ungrammatical ([+licensor, -gap] and [-licensor, +gap]) and the two kinds of sentences that are grammatical ([+licensor, +gap] and [-licensor, -gap]). When this metric is positive, we can conclude that the model reflects some understanding of filler-gap dependencies because it finds ungrammatical sentences as a group more surprising than grammatical ones. A model can score high on this metric by knowing the bijectivity of filler-gap dependencies, i.e. if a sentence has a gap it should have a licensor *and* if a sentence has a licensor it should have a gap. However, a model can achieve a large positive score on this metric even if it only encodes one direction of the bijectivity. For instance, if the presence of a licensor reduces the surprisal of a sentence with a gap, but has no impact on a sentence without a gap, the formula will indicate that the model has learned filler-gap dependencies even though it has learned only a single direction of the dependence. In search for stronger evidence, we propose two criteria: (8) and (9).

(8) **Does surprisal "flip"?**
Is the surprisal higher for [+licensor] than [-licensor] when [-gap] and is it lower for [+licensor] than [-licensor] when [+gap]?

(9) **Does surprisal "divide" by grammaticality?**
Within the four variants of a filler-gap de-

pendency (e.g. (5)), do grammatical variants have lower surprisals than their ungrammatical counterparts?

A flip in surprisal (8) is stronger than a high *wh*-licensing interaction because the former implies the latter but not the other way around. To see this, consider the previous scenario where *wh*-licensing interaction is high despite the LM not having learned bijectivity. Then, surprisal is not higher for [+licensor] when [-gap], so there is no flip. Surprisal will flip only if LMs learn bijectivity.

A division in surprisal by grammaticality (9) is even more demanding than a flip, but it is a reasonable criterion given that probabilistic measures correlate with acceptability judgments (Lau et al., 2014, 2015, 2017). The method of comparing probabilities within minimal pairs has also driven much other work in the assessment of neural networks' understanding of syntax (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018).

For our first study, we assess Wilcox et al.'s LMs' acquisition of filler-gap dependencies by checking for flips (8) and divisions (9) on three kinds of probabilistic metrics calculated from the data from their first experiment, which we describe now.

3.1 Metrics

Local surprisal Wilcox et al. (2018) always measures local surprisal on the post-gap region regardless of [gap]. This means local surprisal for a [-gap] sentence is measured after the filled gap, e.g. in a [-gap] variant of (7-a), the measurement takes place at *at sunrise* rather than *a gazelle*. However, a spike in surprisal due to illicit filled gaps might occur at the filled gap rather than at the post-gap region (Roger Levy, p.c.). Taking this possibility into account, local surprisal is measured at the filled gap for [-gap] sentences in later work such as (Wilcox et al., 2019b). We follow this practice and measure local surprisal at different regions depending on [gap]. Note that we can no longer check for divisions by grammaticality with local surprisal as we cannot compare surprisals between [+gap] and [-gap] conditions, since [gap] perfectly confounds with region. Nevertheless, this allows us to check for surprisal flips, which only depends on comparisons within [+gap] and [-gap].

Global surprisal We follow Wilcox et al. (2018) in measuring global surprisal. We normalize it by region length, which is otherwise an obvious confound – the embedded clause in [+gap] sentences is shorter and thus likely less surprising than [-gap] sentences.

SLOR The syntactic log-odds ratio (SLOR, Pauls and Klein, 2012) for a sentence s is sentence probability normalized for word frequency and word count, and has been shown to positively correlate with human acceptability judgments (Lau et al., 2017).

We train two unigram models on the training sets for the Google model and the Gulordava model respectively with add-one smoothing, and use the unigram model that matches the LM in our calculation of SLOR.

3.2 Experiments

We use mixed-effects models in all analyses. To check if the metrics flip, we predict the metrics with a fixed effect of [+licensor] on [-gap] and [+gap] sentences separately. To check if the metrics divide by grammaticality (9), we predict the metrics with a fixed effect of the grammaticality variable [+gram], defined as [gram] = NOT ([licensor] XOR [gap]), on all data.[2] We always include a random intercept by sentence, not by variant, i.e. all variants in (5) count as the same sentence.

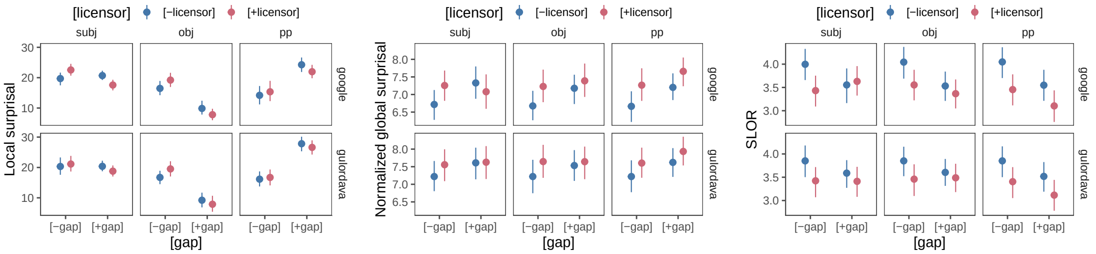
In the plots, points indicate means and error bars indicate 95% confidence intervals thereof computed by non-parametric bootstrapping.

We analyze the data from Wilcox et al. (2018)'s first experiment, which shows that both LMs show positive though different *wh*-licensing interactions for sentences each containing either a subject, object or a prepositional object (PP) gap, indicating that they learn that gaps may occur in all three syntactic positions.

3.3 Local surprisal

Figure 1a shows local surprisal. We see flips in all conditions with significant differences between the [+/-licensor] sentences ($p < 0.05$). This allows us to make a stronger statement about the LMs' acquisition of filler-gap dependencies than Wilcox et al. (2018), whose *wh*-licensing interaction metric can only lead them to conclude that these LMs learn that having a licensor has a different effect on surprisal depending on [gap].

[2] [+gram] iff either [+licensor, +gap] or [-licensor, -gap], i.e. iff the sentence is grammatical insofar as filler-gap dependencies are concerned.



(a) Local surprisal.

(b) Normalized global surprisal.

(c) SLOR.

Figure 1: Local, normalized global surprisals and SLOR for sentences containing subject, object and PP gaps.

3.4 Global surprisal

Figure 1b shows global surprisal. Flips are only observed for the Google model with subject gaps ([+gap]: $\beta = -0.24, p < 0.05$, [-gap]: $\beta = 0.53, p < 0.05$). Elsewhere, [-licensor] is less surprising than [+licensor], and [-gap] less surprising than [+gap].

As for divisions, grammatical sentences are less surprising than ungrammatical sentences in all conditions, but this difference is significant only for subject gaps for both models (Google: $\beta = -0.39, p < 0.05$, Gulordava: $\beta = -0.16, p < 0.05$).

3.5 SLOR

While local and global surprisal are negative log-likelihoods, SLOR is based on positive log-likelihood. Thus, we expect grammatical sentences to have higher SLORs than ungrammatical sentences.

Figure 1c shows SLOR. As with global surprisal, we see SLOR flip only for the Google model with subject gaps ([-gap]: $\beta = 0.08, p < 0.05$, [+gap]: $\beta = -0.56, p < 0.05$). In the remaining conditions, SLOR is higher for [-licensor] than [+licensor] and higher for [-gap] than [+gap].

Grammatical sentences have higher SLOR than ungrammatical sentences in all conditions, although the difference is significant only for subject and object gaps for both models ($p < 0.05$).

Contrary to local surprisal, both global surprisal and SLOR display flips and divisions in very few conditions. We tend to observe flips and divisions, if at all, most often in subject gaps, then in object gaps, and least often in PP/goal gaps. This trend is consistent with Wilcox et al. (2018)'s results with *wh*-licensing interaction.

3.6 Summary of the metrics

Why does local surprisal give us the most optimistic assessment of filler-gap dependency acquisition? We note that global surprisal and SLOR suffer from more confounds and thus may reflect grammaticality less purely than local surprisal. For example, the impact of word frequency as a confound on global surprisal and SLOR is greater than that on local surprisal; grammatical combinations of infrequent words can be more surprising than ungrammatical combinations of frequent words, violating the division criterion (9). [3] SLOR attempts to correct sentence probability by unigram frequency but ignores higher order effects, e.g. the [-licensor] bigram *know that* and the [+licensor] bigram *know who/what/where* can have different frequencies, which can correlate with extraction availability (Liu et al., 2019; Richter and Chaves, 2020), and affect the conditional probabilities of all words that follow in an autoregressive model such as LSTMs, potentially drastically affecting sentence-level probability.

4 Study 2: Other constructions

How well do LMs learn other kinds of filler-gap dependency constructions? We generated six sub-datasets that contain five kinds of filler-gap dependencies: `comp-quant` for comparatives (2-a), `cleft-adj` and `cleft-noun` for *it*-clefts (2-b), `topic` for topicalization (2-b),

[3] A reviewer has pointed out that the three metrics are all confounded by word frequency. This is correct, as surprisal measured at any word is influenced by the frequency of said word as well as all words in its context. However, we believe the impact of this confound on global surprisal and SLOR is greater than that on local surprisal simply because the region at which local surprisal is measured is strictly contained by the region at which global surprisal is measured, which in turn is strictly contained by the region at which SLOR is measured, i.e. the entire sentence. Semantic factors can also affect these metrics in a similar way. Metrics that involve surprisals from more words are more heavily impacted by such confounds.

`embwhq` for embedded *wh*-questions (2-d) and `tough` for *tough*-movements (2-e). Each subdataset contains 1200 sentences, or 4800 variants. Like Wilcox et al. (2018), every sentence has four variants generated from combinations of [gap] and [licensor]. For independent reasons, we generate two datasets for *it*-clefts. We describe dataset construction in more detail in Appendix A.

The gap is always an object gap. The gap-containing clause in each sentence may be separated by 0 – 3 levels of **embedding**, of which there are three types: bridges (10-a), complex NP objects (10-b) and interrogatives (10-c). The latter two types of embeddings induce island effects – namely violations of the Complex NP Constraint and the *Wh*-island Constraint (Ross, 1967) – so each sentence is also specified for **islandhood**; a sentence is an island iff it contains a complex NP object or an interrogative embedding.

(10) a. It was Alex$_i$ that **I think that** you met ___$_i$.
b. *It was Alex$_i$ that **I believed his claim that** you met ___$_i$.
c. *It was Alex$_i$ that **I wondered if** you met ___$_i$.

We then assess the LMs' acquisition of these constructions from different perspectives. As we shall see, the LMs learn different constructions to different extents. Embedded *wh*-questions are learned best across the board and topicalizations are learned the worst. The LMs show mixed acquisition for the remaining constructions, with clefts and comparatives learned generally better than *tough*-movement.

We fit mixed-effects models to predict one of the three metrics with a random intercept by sentence for all analyses. We will describe the fixed effect structure for each analysis. To visualize the modeling results and the inferences they license, we plot model effect estimates along with error bars indicating 95% confidence intervals on those estimates.

4.1 Licensor-gap interaction

We first focus on sentences with no embeddings and look at how [licensor] and [gap] affect surprisal and SLOR. For each combination of construction type and LM, we fit for a fixed effect of [+licensor], [+gap] and their interaction. We are specifically interested in the interaction term. A significant licensor-gap interaction that points in the direction of higher probability, i.e. lower surprisal and higher SLOR, means a LM has learned that [+licensor] has a better effect on surprisal / SLOR when [+gap] than when [-gap]. This way of assessing the acquisition of filler-gap dependencies is roughly the same as looking at Wilcox et al. (2018)'s *wh*-licensing interaction, which they obtain by direct calculation from the data instead of from a statistical model.

Figure 2 shows the licensor-gap interactions. For embedded *wh*-questions, the interaction is always significant in the direction of higher probability for both LMs in all three metrics. The Google model shows a highly significant positive interaction for clefts, comparatives and *tough*-movements for all three metrics. However, the Gulordava model shows a significant interaction for clefts and comparatives only for global surprisal and SLOR, and never for *tough*-movements. Both models showed the least understanding of topicalization, here the expected positive interaction was often significantly negative indicating that licensors increased the surprisal of sentences with gaps.

4.2 Flips

Next, we check how often flips occur on the constructions. We first look at sentences with no embeddings. Figure 3 shows the [+licensor] effects. We see that both LMs show flips for embedded *wh*-questions in all three metrics. Clefts flip only in local surprisal for Google and in global surprisal for both LMs. Comparatives only flip in global surprisals for both LMs. No metrics flip for topicalization and *tough*-movement in any condition.

We then look at sentences with one embedding each. The data thus consist of islands and non-islands. Islandhood affects filler-gap dependencies, which are [+gap, +licensor], but not [-gap, -licensor] sentences. [4] We consider the interac-

[4] We expect probabilistic outputs from a human-like LM to be affected by islandhood for [+gap, +licensor] sentences, not for [-gap, +licensor] sentences. This expectation comes from the acceptabilities of these two types of sentences, as illustrated in (i).

(i) I know { that / *who } you believe [the idea that she beat him in the election].

In contrast, Wilcox et al. (2021) expect islandhood to affect surprisals in both [-gap, +licensor] sentences as well as [+gap, +licensor] sentences. This is in line with a view on human sentence processing that humans do not expect a gap in an island, so the filled gap in (i) is equally surprising with or without an upstream licensor (Fodor, 1983; Freedman and Forster, 1985; Stowe, 1986). There is a rich body of literature concerning the debate on the question of how the human parsing mechanism interacts with grammatical constraints such as island constraints, and we believe it would be an interesting research

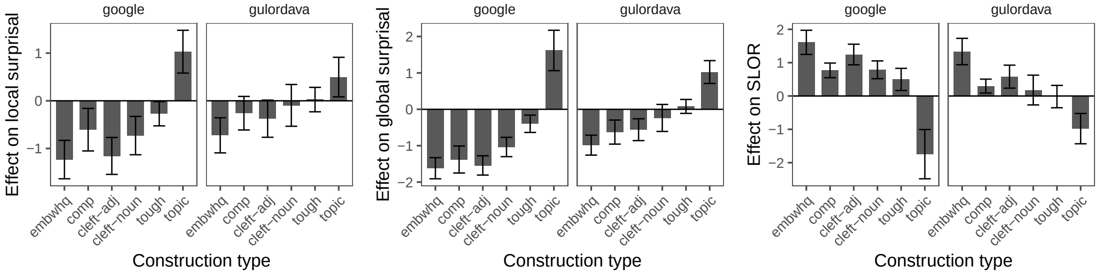


Figure 2: Licensor-gap interaction.

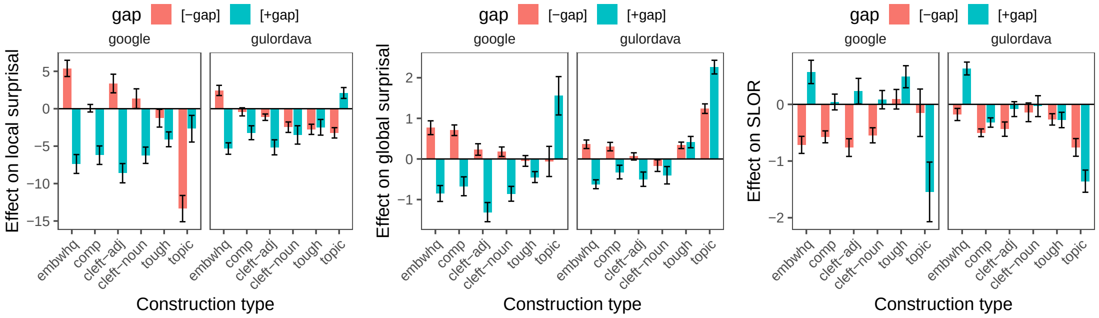


Figure 3: [+licensor] effects on [-gap] and [+gap] sentences.

tion between [licensor] and islandhood for [-gap] and [+gap] sentences separately. Figure 4 shows the [-gap, -licensor, +island] and [+gap, +licensor, +island] effects. When there is neither a licensor or a gap, islandhood does not affect surprisal or SLOR for any of the constructions or either LM. When both are present, islandhood hurts surprisal and SLOR for embedded *wh*-questions for both LMs. Islandhood hurts all three metrics for clefts and comparatives as well except SLOR from the Gulordava model. For *tough*-movement, islandhood does not affect SLOR, but it does associate with lower local surprisal for both LMs and lower global surprisal for the Google model. Islandhood never affects topicalization. From these licensor-islandhood interactions we can conclude that the LMs are learn island constraints to different extents for different constructions.

4.3 Divisions

Finally, we check how often divisions by grammaticality occur on the constructions. We first look at sentences with no embeddings. Figures 5a, 5b show the grammaticality effects. For both LMs, grammatical clefts, comparatives and embedded *wh*-questions have lower global surprisal than their ungrammatical counterparts, whereas for topicalization the grammatical variants are more surprising. Grammatical *tough*-movement is less surprising for Google ($\beta = -0.20, p < 0.05$) but more surprising for Gulordava ($\beta = 0.04, p = 0.55$).

Grammatical clefts, comparatives and embedded *wh*-questions also have higher SLOR, but for Gulordava the difference is non-significant for clefts ($\beta = 0.05, p = 0.48$) and comparatives ($\beta = 0.09, p = 0.24$). Topicalization again is more surprising when grammatical. Grammatical *tough*-movement has slightly higher SLOR for Google ($\beta = 0.20, p = 0.07$) but a non-significant difference for Gulordava ($\beta = -0.005, p = 0.9$).

We then look at all non-island sentences, and consider the interaction between grammaticality and the number of embeddings. Figures 5c, 5d show the interaction effects between grammaticality and the number of embeddings. Increasing number of embeddings is associated with higher global surprisal in all grammatical constructions except topicalization, with non-significant effects in comparatives and *tough*-movement for Gulordava. It is also associated with lower SLOR in the same constructions with non-significant effects in *tough*-movement for both LMs, clefts and comparatives for Gulordava. These patterns suggest that filler-gap dependencies that extend over multiple embeddings are harder for the LMs to process. However,

direction to conduct systematic comparisons between LM and human behaviors in the context of this question (e.g. Wilcox et al. (2019a)).

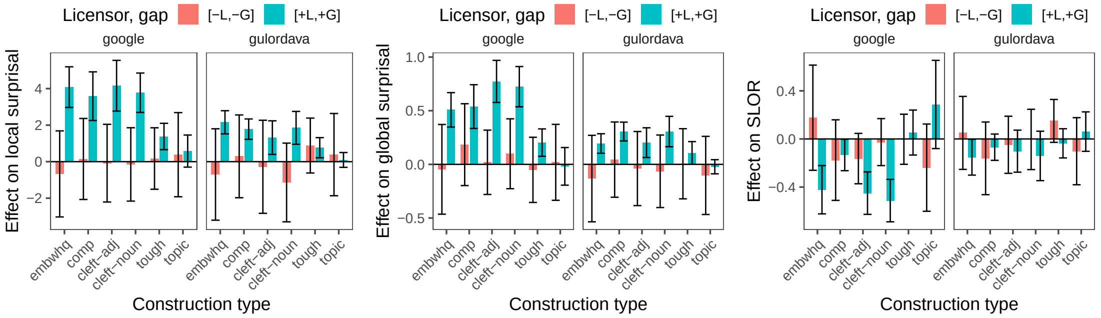
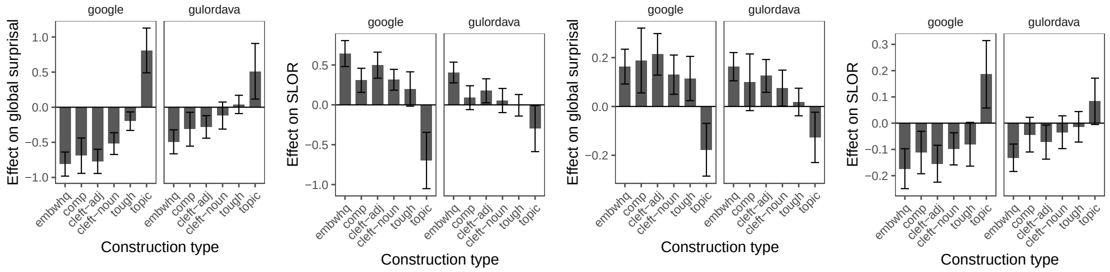


Figure 4: [+island] effects for [-gap, -licensor] and [+gap, +licensor] sentences.



(a) Normalized global surprisal, no embeddings.

(b) SLOR, no embeddings.

(c) Normalized global surprisal, non-islands.

(d) SLOR, non-islands.

Figure 5: Interaction effects between grammaticality and the number of embeddings, fit first on sentences without embeddings then on all non-island sentences.

topicalization has lower surprisal and higher SLOR with more embeddings.

4.4 Summary

We looked at five kinds of filler-gap dependency constructions, and found that the LMs learn different constructions to different extents with respect to licensor-gap interaction, flips, licensor-islandhood interaction, division by grammaticality and the interaction between grammaticality and the number of embeddings. Roughly, the constructions seem to be better learned in the decreasing order of embedded *wh*-questions, clefts and comparatives, *tough*-movement and topicalization.

5 Study 3: Acquisition and frequency

Why do the LMs learn the five filler-gap dependency constructions to different extents? One simple hypothesis is that LMs learn frequent syntactic phenomena better than rare ones (Zhang et al., 2020). To test this, we searched for occurrences of our five constructions in the Brown corpus and the Wall Street Journal corpus from Penn Treebank 3.0 (Marcus et al., 1993) using Tregex (Levy and Andrew, 2006). This gives us an estimate of the relative frequencies of the constructions in a typical written-text corpus, which is what the two LMs were trained on. We choose our licensor-gap interaction from Section 4 to be a quantitative measure of the LMs' acquisition of the constructions. We look for a correlation between licensor-gap interaction and the relative frequency of the constructions; the results are shown in Table 1. We provide the relative frequencies of the constructions and the Tregex scripts we used to search for the constructions in Appendix B.

LM	Metric	r (Brown)	r (WSJ)
Google	global	-0.20	-0.67
	local	-0.32	-0.75
	slor	0.13	0.65
Gulordava	global	-0.32	-0.73
	local	-0.52	-0.82
	slor	0.52	0.86

Table 1: Pearson's r between the licensor-gap interaction and frequency of the filler-gap dependency constructions.

Significance testing is not performed due to the lack of data – there are only five kinds of constructions. The correlation between licensor-gap interaction for the Gulordava model and frequency in the WSJ corpus seems strong, potentially due to a domain similarity between English Wikipedia, which Gulordava was trained on, and WSJ, which consists solely of newspaper articles. The Brown corpus however covers a wider range of older texts. Overall, acquisition of a construction seems to be correlated with its frequency, but more constructions need to be tested in order for the correlation to be non-anecdotal and for this conclusion to be supported.

6 Discussion

We have shown that Wilcox et al. (2018)'s LSTM LMs learn the bijectivity of certain English filler-gap dependency constructions. For embedded *wh*-questions, gaps are less surprising with a licensor than without, and filled gaps are more surprising with a licensor than without. However, this sign of acquisition is stronger for local surprisal than for global surprisal and SLOR. Compared to local surprisal, global surprisal and SLOR are more heavily impacted by confounds such as sentence length and word frequency, which makes the latter two unfair metrics for assessing LMs' syntactic understanding – probability is not all about grammaticality.

As has been correctly pointed out by two reviewers, the connections between probability, categorical grammaticality and gradient acceptability are not innocent. While probability seems to be correlated with acceptability for sentences constructed with round-trip translation, it seems less so with grammaticality for sentences constructed by linguists (Lau et al., 2017; Sprouse et al., 2018). This suggests that probability is a good indicator of unacceptability caused by coarse lexical and syntactic errors introduced by machine translation, but it cannot be used to distinguish between linguist-constructed minimal pairs that often vary very subtly in surface structure. The experimental items in our study are also constructed from a linguistic standpoint. With this in mind, the failure of global and SLOR to indicate correspondence with grammaticality supports the present claim in the literature.

We also see that the LMs learn different filler-gap dependency constructions to different extents, in terms of licensor-gap interaction, flips and divisions, as well as islandhood and the number of embeddings. Moreover, the more frequent a construction is in written English, the more the licensor-gap interaction improves the probability of a filler-gap dependency. While this does not tell us much about what the neural networks have learned, this is a human-like behavior in that frequency affects human language acquisition (Ambridge et al., 2015) and sentence processing (Ellis, 2002).

A systematic investigation with a wider coverage of filler-gap dependency constructions is in order. In this study, we were able to adopt Wilcox et al. (2018)'s 2x2 design because for each of our five constructions, we could either take the filler to be the licensor, or find a construction with a minimal surface difference that does not license gaps, and take that to be our [-licensor] variant. For example, we construct [-licensor] variants of comparatives (... *than* ...) by turning them into coordinate structures (... *and* ...). In other filler-gap dependency constructions, this is much more challenging. For example, infinitival relative clauses (11) are filler-gap dependencies, but it is not obvious how to construct [-licensor] variants for them.

(11) a. Here are **some options**$_i$ for you to choose from ___$_i$.
b. She was the first **person**$_i$ ___$_i$ to point out the mistake.

The experimental paradigm needs to be revised in a way to cover such constructions as well. In future research, we wish to collect human acceptability judgments for our data, and compare our results with the probabilistic outputs from LMs to check the connection between probability and acceptability.

We provide our data and code at `https://github.com/ikazos/scil2022-fgd`. We also thank Roger Levy and a reviewer for pointing out SyntaxGym (Gauthier et al., 2020) to us, which we plan to contribute our data to in the near future.

References

Ben Ambridge, Evan Kidd, Caroline F. Rowland, and Anna L. Theakston. 2015. The ubiquity of frequency effects in first language acquisition. *Journal of Child Language*, 42(2):239–273.

Jean-Philippe Bernardy and Shalom Lappin. 2017. Using deep neural networks to learn syntactic agreement. *Linguistic Issues in Language Technology*.

Rui Chaves. 2020. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 1–11, New York, New York. Association for Computational Linguistics.

Rui P. Chaves and Michael T. Putnam. 2021. *Unbounded Dependency Constructions: Theoretical and Experimental Perspectives*, volume 10. Oxford University Press, USA.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Noam Chomsky. 1977. On Wh-Movement. In Peter W. Culicover, Thomas Wasow, Adrian Akmajian, et al., editors, *Formal Syntax*, pages 71–132.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. Rnn simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th international conference on computational linguistics*, pages 133–144.

Jillian Da Costa and Rui Chaves. 2020. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 12–21, New York, New York. Association for Computational Linguistics.

Nick C Ellis. 2002. Frequency effects in language processing: A review with implications for theories of implicit and explicit language acquisition. *Studies in second language acquisition*, 24(2):143–188.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Janet Dean Fodor. 1983. Phrase structure parsing and the island constraints. *Linguistics and Philosophy*, 6(2):163–223.

Sandra E Freedman and Kenneth I Forster. 1985. The psychological status of overgenerated sentences. *Cognition*, 19(2):101–131.

Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. Rnns as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Ralph Grishman, Catherine Macleod, and Adam Meyers. 1994. Comlex syntax: Building a computational lexicon. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. *arXiv preprint arXiv:1803.11138*.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of lstms to understand negative polarity items. *arXiv preprint arXiv:1808.10627*.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2014. Measuring gradience in speakers' grammaticality judgements. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2015. Unsupervised prediction of acceptability judgements. 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 1618–1628, Beijing, China. Association for Computational Linguistics.

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.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: Tools for querying and manipulating tree data structures. In *LREC*, pages 2231–2234. Citeseer.

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.

Yingtong Liu, Rachel Ryskin, Richard Futrell, and Edward Gibson. 2019. Verb frequency explains the unacceptability of factive and manner-of-speaking islands in english. In *CogSci*, pages 685–691.

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

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

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, Jeju Island, Korea. Association for Computational Linguistics.

Stephanie Richter and Rui Chaves. 2020. Investigating the role of verb frequency in factive and manner-of-speaking islands. In *CogSci*.

John Robert Ross. 1967. *Constraints on variables in syntax*. Ph.D. thesis, Cambridge, MA.

Ivan A. Sag. 2010. English filler-gap constructions. *Language*, 86(3):486–545.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Jon Sprouse, Beracah Yankama, Sagar Indurkhya, Sandiway Fong, and Robert C Berwick. 2018. Colorless green ideas do sleep furiously: gradient acceptability and the nature of the grammar. *The Linguistic Review*, 35(3):575–599.

Laurie A Stowe. 1986. Parsing wh-constructions: Evidence for on-line gap location. *Language and cognitive processes*, 1(3):227–245.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Ethan Wilcox, Richard Futrell, and Roger Levy. 2021. Using computational models to test syntactic learnability. *lingbuzz/006327*.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019a. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 181–190, Florence, Italy. Association for Computational Linguistics.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019b. What syntactic structures block dependencies in rnn language models? *arXiv preprint arXiv:1905.10431*.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R. Bowman. 2020. When do you need billions of words of pretraining data?

A Dataset construction

In this section, we discuss how we constructed the [-licensor] and [-gap] variants of the 5 filler-gap dependency constructions.

[-gap] variants are always created by filling the gap with the filler. The creation of [-licensor] differs over constructions.

A.1 Comparatives

We look at comparative constructions with a comparative quantifier *more* modifying an object NP with *than* leading a subordinating clause containing the gap. We consider the filler to be *a lot of* + matrix object NP. For the [-licensor] variant, we replace *more* with *a lot of* and replace *than* with *and*, giving us a coordinate structure that does not license a gap.

(12) a. [+licensor, +gap]
Mary bought **more books** this month **than** John bought ___ last month.

b. [+licensor, -gap]
*Mary bought **more books** this month **than** John bought **a lot of books** last month.

c. [-licensor, +gap]
*Mary bought **a lot of books** this month **and** John bought ___ last month.

d. [-licensor, -gap]
Mary bought **a lot of books** this month **and** John bought **a lot of books** last month.

A.2 Clefts

We look at clefts with object gaps of the structure *It was* ... *that* .... The two subdatasets `cleft-adj` and `cleft-noun` are created with different strategies for creating the [-licensor] variants. In `cleft-adj`, the [-licensor] variants are created by replacing the filler with an adjective that take an extraposed sentential subject, e.g. *apparent*. In `cleft-noun`, the [-licensor] variants are created by replacing the filler with a noun that take an extraposed sentential subject, e.g. *a fact*. We collected a list of such adjectives and nouns from COMLEX Syntax (Grishman et al., 1994).

(13) a. [+licensor, +gap]
It was **books** that Mary bought ___ last month.
b. [+licensor, -gap]
*It was **books** that Mary bought **books** last month.
c. [-licensor, +gap] (`cleft-adj`)
*It was **apparent** that Mary bought ___ last month.
d. [-licensor, +gap] (`cleft-noun`)
*It was **a fact** that Mary bought ___ last month.
e. [-licensor, -gap] (`cleft-adj`)
It was **apparent** that Mary bought **books** last month.
f. [-licensor, -gap] (`cleft-noun`)
It was **a fact** that Mary bought **books** last month.

A.3 Embedded *wh*-questions

We look at embedded *wh*-questions with object gaps. The matrix verb selects for either a sentential or an interrogative complement; we gathered a list of such verbs from VerbNet (Schuler, 2005) and from Wilcox et al. (2018)'s data. Following Wilcox et al. (2018), we replace the *wh*-phrase leading the interrogative complement with *that* for the [-licensor] variants.

(14) a. [+licensor, +gap]
Clara knows **what** Mary bought ___ last month.
b. [+licensor, -gap]
*Clara knows **what** Mary bought **books** last month.
c. [-licensor, +gap]
*Clara knows **that** Mary bought ___ last month.
d. [-licensor, -gap]
Clara knows **that** Mary bought **books** last month.

A.4 Topicalization

We look at topicalization (also known as complement preposing (Huddleston and Pullum, 2002)) with object gaps. Unlike the other constructions, topicalization does not allow subject gaps – this is one of the reasons why we exclusively generate object gaps throughout all constructions. The filler is always a definite NP, which helps with a focus interpretation. For the [-licensor] variants, we simply delete the filler and the comma. For the [-gap] variants, we fill the gap with the filler directly instead of e.g. a referential pronoun, because that would give us left-dislocation for [+licensor, -gap], which is a grammatical construction.

(15) a. [+licensor, +gap]
These books, Mary bought ___ last month.
b. [+licensor, -gap]
*__These books__, Mary bought **these books** last month.
c. [-licensor, +gap]
*Mary bought ___ last month.
d. [-licensor, -gap]
Mary bought **these books** last month.

A.5 *Tough*-movement

We look at *tough*-movement with object gaps. We select matrix adjectives that license hollow *to*-infinitivals (Huddleston and Pullum, 2002). For the [-licensor] variants, we replace the filler with *it*.

(16) a. [+licensor, +gap]
These books are impossible to finish ___ in a day.
b. [+licensor, -gap]
*__These books__ are impossible to finish **these books** in a day.
c. [-licensor, +gap]
*__It__ is impossible to finish ___ in a day.
d. [-licensor, -gap]
It is impossible to finish **these books** in a day.

B Data and Tregex scripts for Study 3

The relative frequencies for each construction type in the Brown corpus and the WSJ corpus are listed in Table 2. Here are the Tregex scripts used to search for the occurrences for each construction.

B.1 Clefts

This covers both `cleft-adj` and `cleft-noun`. The script is: `S-CLF`

Construction	Freq (in Brown)	Freq (in WSJ)
Clefts	108	65
comp-quant	9	41
embwhq	280	146
topic	119	14
tough	36	79

Table 2: Relative frequency for each construction type in the Brown corpus and the WSJ corpus.

B.2 Comparatives

This covers comp-quant. The script is: @NP << more & !<< (@ADJP << more) < (PP|SBAR < (__ < than) & < @S)

B.3 Embedded *wh*-questions

This covers embwhq. The script is: VP < (SBAR < (/WH*/ << what|who))

B.4 Topicalization

This covers topic. We first look for occurrences of /NP-TPC-?/ !<< ``, then subtract the number of occurrences of `` $+ (/NP-TPC-?/ !<< ``) to rule out false positives.

B.5 *Tough*-movement

This covers tough. The script is: ADJP-PRD < (SBAR < /WHNP-*/).

Inferring Inferences: Relational Propositions for Argument Mining

Andrew Potter
Computer Science & Information Systems Department
University of North Alabama
Florence, Alabama, USA
apotter1@una.edu

Abstract

Inferential reasoning is an essential feature of argumentation. Therefore, a method for mining discourse for inferential structures would be of value for argument analysis and assessment. The logic of relational propositions is a procedure for rendering texts as expressions in propositional logic directly from their rhetorical structures. From rhetorical structures, relational propositions are defined, and from these propositions, logical expressions are then generated. There are, however, unsettled issues associated with Rhetorical Structure Theory (RST), some of which are problematic for inference mining. This paper takes a deep dive into some of these issues, with the aim of elucidating the problems and providing guidance for how they may be resolved.

1 Introduction

The logic of relational propositions is a process for rendering texts as expressions in propositional logic. It is based on Rhetorical Structure Theory, as defined by Mann and Thompson (1986, 1988). From rhetorical structures, relational propositions are defined, and from relational propositions, logical expressions are constructed (Potter, 2019). Inferences contained in these expressions tend to be applicable to their respective texts as well, with little to no loss of coherence (Potter, 2021). This is significant for argument mining, as it suggests that RST can be used to identify the inferential structure of discourse. When combined with systems for automated identification of RST structures, this would provide for an end-to-end process for discovering generalized inferential structures in free texts.

There are, however, some limitations and unsettled issues associated with this. Multiple annotation guidelines have been developed for use in performing RST analyses. The relation set for which logical generalization has been defined is the extended Mann and Thompson set.[1] Because the relational definitions provided with these guidelines focus on the intended (rhetorical) effect of the relations, they appear to be well-suited for inference mining (Potter, 2019, 2020). However, these definitions are necessarily reliant on analyst intuition, and this makes the annotation task cumbersome, particularly for large corpuses. The annotation guidelines developed by Carlson and Marcu (2001) are generally more reliant on syntactical features, and therefore more suitable for automated analysis, and yet less appropriate for inference mining. Closer to Mann and Thompson's approach are the guidelines defined by Stede et al. (2017). These provide both rhetorical and syntactic definitions, and a relation set somewhat larger than Mann and Thompson's, but much smaller than that of Carlson and Marcu. Among these sets not only are there differences in relation identification, but their definitions are sometimes inconsistent with one another. Even within the fundamentals of segmentation there are differences of opinion as to what constitutes a discourse unit.

There have been several efforts to clarify such issues (e.g., Nicholas, 1994; Stede, 2008; Wan, Kutschbach, Lüdeling, & Stede, 2019), but debate and disagreement continue unabated. Ultimately, if Rhetorical Structure Theory is to be used for inference mining, or indeed it is to continue to distinguish itself among theories of coherence relations, not only is a stable relation set needed,

[1] https://www.sfu.ca/rst/01intro/definitions.html

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 89-100.
Held on-line February 7-9, 2022

syntactically defined relations must be subsumable by their pragmatic and semantic counterparts, and their inferential characteristics must be clearly specified.

Although realization of these goals is beyond the scope of this paper, progress can be made through close examination and explication of some of the problematic relations. And this will shed light on how other issues might be approached. I have selected CIRCUMSTANCE, SOLUTIONHOOD, and ELABORATION for this analysis. While these relations are not alone in their need for attention, the issues associated with them are fundamental, not just with respect to their inferentiality, but to RST in general. CIRCUMSTANCE has been subject to various interpretations and definitions, and these have implications for when it should be used, what qualifies as a segment for these relations, and what inferences may be drawn from it. Stede et al. (2017) deprecated it in favor of the BACKGROUND relation, and similarly Carlson and Marcu (2001) classified it as a background subtype. Similarly, SOLUTIONHOOD can be difficult to distinguish from PREPARATION (Zeldes & Liu, 2020), and while defined by both Mann and Thompson and Stede, et al., it has no direct counterpart in Carlson and Marcu. ELABORATION is among the most frequently used of relations, and yet while there seems to be general agreement that its satellite will contain additional information about its nucleus, specifics as to how it should be defined diverge from there, with Mann and Thompson identifying six different ways this can occur, Stede et al., splitting the relation in two, and Carlson and Marcu subdividing it into eight separate relations.

Moreover, much has been made about the distinction between presentational (pragmatic) and subject matter (semantic) relations. This distinction was introduced somewhat tentatively by Mann and Thompson, but has been treated as gospel ever since. Claims such as those of Azar (1995, 1997, 1999) that only selected relations among the presentationals can be construed as argumentative have only served to strengthen this view. And while subject matter relations may not be interpersonal in the sense found in some of the presentational relations, this examination of CIRCUMSTANCE, SOLUTIONHOOD, and ELABORATION will show they are both inferential and instrumentally argumentative.

The paper is organized as follows. The next section provides a brief review of related research. This is followed by an overview of the theoretical background for this study. Next, I examine the selected RST relations and their associated issues from the perspective of the logic of relational propositions with the aim of clarifying their inferential features. The paper closes with a discussion and summary of the results.

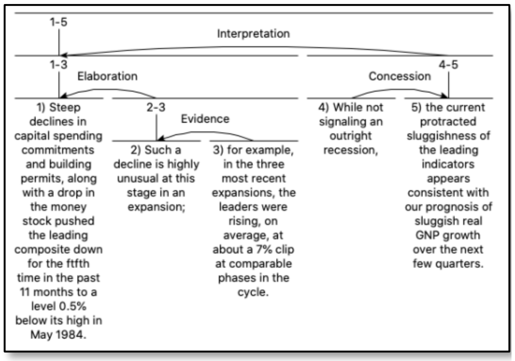

1-5
Interpretation
1-3
Elaboration
4-5
Concession
1) Steep declines in capital spending commitments and building permits, along with a drop in the money stock pushed the leading composite down for the ftfth time in the past 11 months to a level 0.5% below its high in May 1984.
2-3
Evidence
2) Such a decline is highly unusual at this stage in an expansion;
3) for example, in the three most recent expansions, the leaders were rising, on average, at about a 7% clip at comparable phases in the cycle.
4) While not signaling an outright recession,
5) the current protracted sluggishness of the leading indicators appears consistent with our prognosis of sluggish real GNP growth over the next few quarters.

Figure 1: Nested RST Structures (Mann & Thompson, 2000)

2 Related work

Studies in the relationship between RST and argumentation are numerous (e.g., Abelen, Redeker, & Thompson, 1993; Azar, 1995, 1997; Doronkina, 2017; Galitsky, Ilvovsky, & Kuznetsov, 2018; Garcia-Villalba & Saint-Dizier, 2012; Green, 2010; Imaz & Iruskieta, 2017; Mitrović, O'Reilly, Mladenović, & Handschuh, 2017; Musi, Ghosh, & Muresan, 2018; Peldszus, 2016; Peldszus & Stede, 2013; Rocci, 2021; Stede, 2020; Wyner & Schneider, 2012), but without focus on the inferential characteristics of rhetorical relations.

On the other hand, the relationship between coherence relations and logic has also been the subject of extensive study (e.g., Asher & Lascarides, 2003; Danlos, 2008; González & Ribas, 2008; Groenendijk, 2009; Hobbs, 1979, 1985; Marcu, 2000; Potter, 2019, 2020, 2021; Sanders, Spooren, & Noordman, 1992; Wong, 1986). Inevitably, the question arises: Why not Segmented Discourse Representation Theory? SDRT builds on the perspective of discourse as a dynamic phenomenon in which context is modified with each successive segment (Asher & Lascarides, 2003), and discourse structure is viewed as a systematic extension of truth-conditional semantics, whereas intentionality is fundamental to the RST perspective. That this intentionality lends itself to argumentative, interpretation is apparent (Potter, 2007, 2008a, 2008b, 2009, 2010, 2019, 2020, 2021), and as such is of interest to argument mining. So far as I know, other than these studies, there have been no attempts to establish a specific alignment between RST and propositional logic.

3 Theoretical Background

The fundamental bases for this work are Rhetorical Structure Theory (RST), relational propositions, and the logic of relational propositions. Rhetorical Structure Theory (RST) is a theory of text organization (Mann & Thompson, 1988). It is used for analyzing texts in terms of the relations that hold among its discourse units. A relation consists of three parts: a satellite, a nucleus, and a relation. Figure 1 shows an RST analysis containing five discourse units related by the ELABORATION, EVIDENCE, CONCESSION, and INTERPRETATION relations. The distinction between satellite and nucleus arises as a result of the asymmetry of the relations. Within a relation, the nucleus is more central to the writer's purpose than the satellite. Thus unit 3 is the satellite of 2, and span [2-3] is the satellite of 1. Unit 4 is the satellite of 5, and [4-5] is the satellite of [1-3]. A defining characteristic of RST relations is their intended effects. Each relation has a defined effect, representing the writer's intention for the relation, as determined by the RST analyst. For example the intended effect of the EVIDENCE relation is *acceptance* of the situation presented in the nucleus. This effects field is what makes RST *rhetorical*.

Relational propositions provide a propositional analog to RST structures, with relations being expressed as propositions. These propositions are implicit assertions occurring between clauses in a text and are essential to the effective functioning of the text (Mann & Thompson, 1986). A relational proposition consists of a relation (or predicate) and two variables, one of which corresponds to the RST satellite and the other to the nucleus. Complex relational propositions can be expressed using a predicate notation defined by Potter (2019). This supports the representation of complex RST structures in compact functional form. However, conceptualizing RST analyses as relational propositions provides more than a space-saving alternative to RST diagrams. Relational propositions provide a means for exploring texts as inferential structures, and this in turn sheds light on the nature of nuclearity and its role in discourse coherence—and as a side-effect, it exposes some issues in RST, hence providing motivation for the investigation that led to the writing of this paper.

As developed by Potter (2019, 2021), each of the RST relations supports a logical interpretation, and most of these interpretations are not only inferential, and but tautological as well, which is to say, they implement a valid rule of inference. This can best be explained by way of example. The relational proposition for the EVIDENCE relation shown in Figure 1 is *evidence(3,2)*. The intended effect is that the satellite, unit *3*, provides evidence in support of *2*. In the analysts' estimation, the reader might not believe *2* without the supporting evidence provided by *3* So *3* gives credence to *2*, leading to acceptance of *2*. Thus the logical form of *evidence(s, n)* is realized as modus ponens, $(((s \rightarrow n) \wedge s) \rightarrow n)$. The relation is not merely conditional, since *s* is asserted. Further, in the example, the relational proposition, *evidence(3,2)*, is positioned as the satellite to unit *1*, resulting in the relational proposition *elaboration(evidence(3,2),1)*. As argued below in Section 6, *elaboration* is inferential insofar as the satellite supports the reader's comprehension of the nucleus by providing additional information. Like *evidence*, the logical form of *elaboration* is modus ponens, $(((s \rightarrow n) \wedge s) \rightarrow n)$. A couple of important things are happening here. The first is that the *evidence* modus ponens has been nested within the *elaboration* modus ponens, functioning as its satellite, and as a premise in the *elaboration* argument. So there is an inferential dependency of one upon the other, such that if fully realized, we have one valid argument as premise to another, i.e., a tautology within a tautology:

$$((((((3 \rightarrow 2) \wedge 3) \rightarrow 2) \rightarrow 1) \wedge$$
$$(((3 \rightarrow 2) \wedge 3) \rightarrow 2)) \rightarrow 1)$$

The second point has to do with the integration of the Boolean domains of these tautologies. As thus far analyzed, the text consists of one argument that establishes acceptance, and this acceptance is then used in a second argument to establish comprehension. This relational proposition, *elaboration(evidence(3,2),1)*, is then used as the nucleus of an *interpretation* predicate. Like *elaboration*, the *interpretation* predicate supports comprehension, yet not by extending the subject matter, but rather by introducing an additional conceptual framework, taking the subject matter to another level. So the polarity of inference is reversed. As INTERPRETATION is defined, its nucleus is leveraged to support the satellite: $(((n \to s) \land n) \to s)$. At this point, things begin to get a little more interesting. The *s* of the interpretation is also the *n* of a *concession*. With *concession*, the writer acknowledges the situation presented in the satellite but asserts that, although there might seem to be an incompatibility between the satellite and the nucleus, the satellite and nucleus are compatible. The writer holds the nucleus in positive regard, and by indicating a lack of incompatibility with the satellite, seeks to increase the reader's positive regard for the nucleus (Potter, 2019; Thompson & Mann, 1986). In other words, while the satellite might seem to indicate rejection of the nucleus, it does not do so, and recognition of their compatibility increases acceptance of the nucleus. In other words, since the satellite does not imply the negation of the nucleus, the nucleus holds:

$$(((\neg(s \to \neg n) \to n) \land \neg(s \to \neg n)) \to n)$$

This is the logical structure of the nucleus of the *interpretation* relational proposition. It may seem a bit complicated, but it is actually just an odd modus ponens. So the full relational proposition of the text under examination is

interpretation(concession(4,5),
elaboration(evidence(3,2),1))

which expands to the logical expression:

$$((((((((3 \to 2) \land 3) \to 2) \to 1) \land (((3 \to 2) \land 3) \to 2)) \to 1) \to (((\neg(4 \to \neg 5) \to 5) \land \neg(4 \to \neg 5)) \to 5)) \land ((((((3 \to 2) \land 3) \to 2) \to 1) \land (((3 \to 2) \land 3) \to 2)) \to 1)) \to (((\neg(4 \to \neg 5) \to 5) \land \neg(4 \to \neg 5)) \to 5))$$

The *evidence* and *concession* predicates specify a Boolean domain of acceptance, *elaboration* of clarification, *interpretation* of illumination or insight, and the overarching logic is one of coherence rather than truth. Thus the logic of discursive coherence subsumes the Boolean domains of rhetorical relations. For each of the relations examined here, identification of the Boolean domain plays an important role.

4 Circumscribing CIRCUMSTANCE

In CIRCUMSTANCE, as defined by Mann and Thompson (1988), the satellite sets a framework for the subject matter within which the reader is intended to interpret the nucleus. Carlson and Marcu (2001) and Stede et al. (2017) concur in this definition. The inferentiality of CIRCUMSTANCE is supported in part by its affinity with the BACKGROUND relation. Stede et al. (2017) argued that the BACKGROUND relation should usually be preferred over CIRCUMSTANCE, because in their view, BACKGROUND is more informative. Potter (2019) defined CIRCUMSTANCE as a causal relation, with possible presentational features. Carlson and Marcu (2001) recognized the similarity between CIRCUMSTANCE and BACKGROUND, but noted that CIRCUMSTANCE tends to be more clearly delimited, and as such is stronger than BACKGROUND. As a pragmatic relation, the inferentiality of BACKGROUND, with a Boolean domain of comprehensibility, is well substantiated. To the extent that CIRCUMSTANCE presents as a special case of BACKGROUND, it too can be expected to be inferential. But I believe the case for inferentiality for CIRCUMSTANCE can stand on its own.

CIRCUMSTANCE is categorized as a subject matter or semantic relation. That a relation might be designated as semantic indicates that since reader acceptance is presumed, this might seem to suggest no inferential activity would obtain. And yet the asymmetry of semantic relations indicates otherwise: these are not merely conjunctions. That their intended effect may be realized without persuasion does not eliminate the necessity for reasoning. The intended effect of each of the semantic relations is predicated on a Boolean domain, albeit more subtle than one of truth and falsity. In CIRCUMSTANCE, the inferential feature, while subtle, is more pronounced than in some others. Its Boolean domain is a delimitation of context. The satellite circumscribes the universe of discourse within which the nucleus holds. In some cases, the strength of the circumscription may be sufficient that the designation of argumentative would be warranted. Figure 2 shows an example of CIRCUMSTANCE. The text is from US President Ronald Reagan's first inaugural address. The nucleus is perhaps one of his most famous quotations. Giving it some context should be helpful both in clarifying what he really said and in explicating the CIRCUMSTANCE relation. The crisis

Reagan had in mind is described in some detail earlier in the address, that inflation is the worst in American history, that unemployment is high, that taxes and the deficit are too high, that personal freedoms have been curtailed, and in general things are tough all over. These are the circumstances under which we are asked to accept that government is not the solution to our problems, that government is the problem.

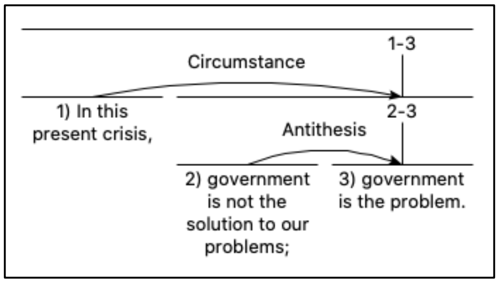

1-3
Circumstance
1) In this present crisis,
2-3
Antithesis
2) government is not the solution to our problems;
3) government is the problem.

Figure 2: CIRCUMSTANCE as Delimiting

The satellite of CIRCUMSTANCE circumscribes the conditions under which the nucleus holds. In this sense it is similar to CONDITION relation, except that the CONDITION satellite expresses a hypothetical, future, or otherwise unrealized situation, whereas with CIRCUMSTANCE the satellite is not unrealized. Thus while CONDITION is $(s \rightarrow n)$, CIRCUMSTANCE is $(((s \rightarrow n) \land s) \rightarrow n)$, or *modus ponens*. Note that this differs from Potter's (2019) description of the logic of CIRCUMSTANCE as subject matter implicative: $(s \land (n \land (s \rightarrow n)))$. That definition was intended to account for the relation as a semantic rather pragmatic, property of the relation. However, this is unnecessary and redundant and moreover an inaccurate rendering of the relation. The Boolean domain of CIRCUMSTANCE is one of contextual enablement. Anything outside the circumscribed universe of discourse cannot be presumed to hold. If the crisis Reagan alluded to no longer persists, then the role of government in solving problems may be in good stead. And a logical analysis of the text bears this out. The relational proposition for the RST analysis shown in Figure 2 is *circumstance(1,antithesis(2,3))*. With *antithesis*, the intended effect is to increase the reader's positive regard for the situation presented in the nucleus. The satellite is incompatible with the nucleus, such that the reader cannot have positive regard for both the nucleus and the satellite: *….government is not the solution to our problems; [on the contrary,] government is the problem*. This incompatibility increases the reader's positive regard for the nucleus. The satellite of the CIRCUMSTANCE relation identifies an enabling context through which the situation identified in the nuclear proposition is realized. Its logical analog as this nested *modus ponens*:

$$\begin{gathered}(((1 \rightarrow (((2 \lor 3) \land \neg 2) \rightarrow 3)) \land 1) \rightarrow \\ (((2 \lor 3) \land \neg 2) \rightarrow 3))\end{gathered}$$

Lest there be any doubt as to the implicativeness of the relation between the CIRCUMSTANCE satellite and the disjunctive syllogism, consider the effect of eliminating the satellite. The relational proposition is then *antithesis(2,3)*, or $(((2 \lor 3) \land \neg 2) \rightarrow 3)$, with the corresponding text making a general claim about the inefficacy of government. This is not what Reagan said. In specifying an enabling context, the CIRCUMSTANCE relation delimits the scope of the situation identified by the nucleus. Anything beyond that scope is unspecified. Taking the nucleus out if its circumstantial context is to remove support for writer's claim.

As a valid argument within a valid argument, the logical expression is a tautology, that is to say, the expression $(((1 \rightarrow (((2 \lor 3) \land \neg 2) \rightarrow 3)) \land 1) \rightarrow (((2 \lor 3) \land \neg 2) \rightarrow 3))$ is true for all possible values of *1*, *2*, and *3*. That these expressions are tautologies should be of no concern so long as it is the coherence of the text that is of interest. That is, the logical definitions are based on a presumed realization of the writer's intended effect. While this is consistent with the expectation of coherence, for critical assessment, the presumed realization of intended effect amounts to begging the question. It is the soundness of the expression that is of interest. For an examination of soundness, it is the premises, and not what may be inferred from them, that is of interest. The premises of the argument are found, as expected, in the left hand side (LHS) of the logical expression, to the left of the outermost implication. Restating the logical expression,

$$\begin{gathered}(((1 \rightarrow (((2 \lor 3) \land \neg 2) \rightarrow 3)) \land 1) \rightarrow \\ (((2 \lor 3) \land \neg 2) \rightarrow 3))\end{gathered}$$

using only the LHS for each relation reduces to

$$((1 \rightarrow ((2 \lor 3) \land \neg 2)) \land 1)$$

Although negation of the LHS will not necessarily negate the RHS, an LHS once negated deprives the discourse of its intended effect. Since the burden of persuasion is on the LHS, negation is sufficient for rejection of the RHS. More specifically, negation of the CIRCUMSTANCE satellite is sufficient to imply negation of the LHS of the expression, such that $(\neg 1 \rightarrow \neg((1 \rightarrow ((2 \lor 3) \land \neg 2)) \land 1))$ is a valid argument. Thus one might challenge the claim that there was any crisis, e.g., arguing that the claim of crisis was a politically motivated fabrication, and this would suffice to weaken the generalization that government is the problem. Of further interest is the possibility of inferring 3 directly from 1. As it happens, $(((1 \rightarrow ((2 \lor 3) \land \neg 2)) \land 1) \rightarrow (1 \rightarrow 3))$ and $(((1 \rightarrow ((2 \lor 3) \land \neg 2)) \land 1) \rightarrow 3)$ are both valid. This shows that the satellite of the CIRCUMSTANCE relation enjoys a transitive relationship with the nucleus of the ANTITHESIS relation. That is to say, the reduced text

> *1) In this present crisis,*
> *3) government is the problem.*

is both logical and plausibly coherent. That this abridgement of the inferential path is readable, despite the loss of rhetorical force, lends support to the inferential interpretation of the CIRCUMSTANCE relation.

5 SOLUTIONHOOD and its subtypes

SOLUTIONHOOD as defined by Mann and Thompson the satellite presents a problem and the nucleus constitutes a solution. *Problem* as used here is broadly defined, and may be presented as a question, a request, a description of a desire or goal, or any one of a variety of other similar situations. The use of the interrogative in discourse, when the writer raises the question and follows it with a response, is quite different from raising a question and leaving it for the reader to answer. In this respect SOLUTIONHOOD is akin to the PREPARATION relation, but more focused. The satellite provides the setup or prompt for presenting the nucleus. The question or problem can be treated as a propositional function, or specification of a query (Hintikka, 2007). Solutions and answers provide the information needed to resolve the problem. From the query, the solution is instantiated (Potter, 2019). It is in this sense that the nucleus is inferred from the satellite, or that the answer follows coherently from the question.

Instantiations of SOLUTIONHOOD may be simple, e.g., *I'm hungry. Let's go to the Fuji Gardens*, where the speaker's announcement of hunger is given as sufficient reason for dinner at a Japanese restaurant. Or they may be complex. Figure 3 shows the satellite of a SOLUTIONHOOD relation in use as the setup for an extended argument. The argument is from Gilbert Ryle's philosophical treatise, *The Concept of Mind*. Through multiple layers of EVIDENCE, the writer positions the argument as a response to the question posed in the satellite, but in going beyond simply satisfying the question, it seeks to affirm an assumption implicit in the question, that people are strongly drawn to believe the thesis of mind-body dualism. A case of complex question perhaps? Ryle could have begun with an assertion rather than a

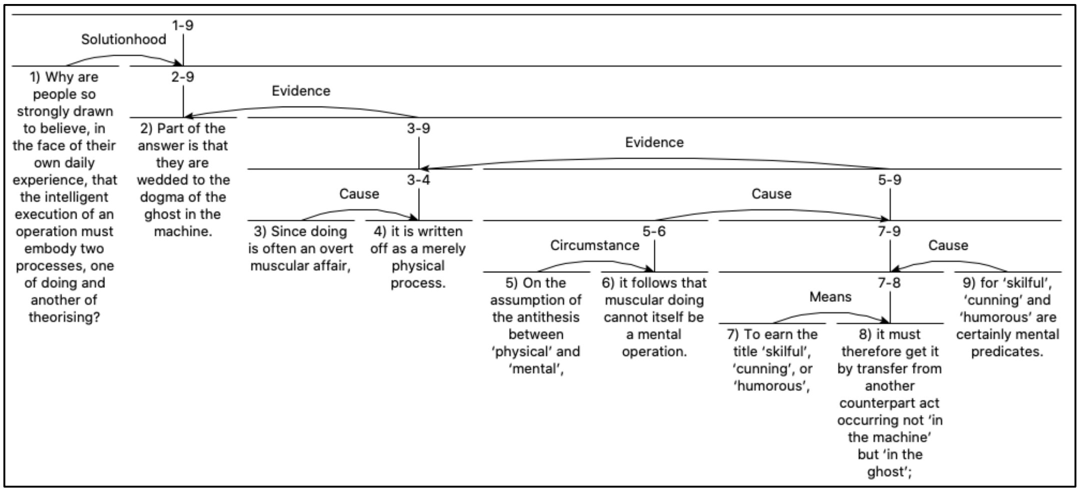

1-9
Solutionhood
2-9
Evidence
1) Why are people so strongly drawn to believe, in the face of their own daily experience, that the intelligent execution of an operation must embody two processes, one of doing and another of theorising?
2) Part of the answer is that they are wedded to the dogma of the ghost in the machine.
3-9
Evidence
3-4
Cause
5-9
Cause
3) Since doing is often an overt muscular affair,
4) it is written off as a merely physical process.
5-6
Circumstance
7-9
Cause
5) On the assumption of the antithesis between 'physical' and 'mental',
6) it follows that muscular doing cannot itself be a mental operation.
7-8
Means
9) for 'skilful', 'cunning' and 'humorous' are certainly mental predicates.
7) To earn the title 'skilful', 'cunning', or 'humorous',
8) it must therefore get it by transfer from another counterpart act occurring not 'in the machine' but 'in the ghost';

Figure 3: SOLUTIONHOOD as Setup for an Extended Argument (Ryle, 1949)

question (e.g., “People are strongly drawn to believe…”) followed by the support provided in span 2-9. But this would have shifted the locus of effect from segment *2* to segment *1*. That is not the claim he sought to establish. That people ascribe to the dogma of the ghost in the machine is a core thesis of his critique of René Descartes. SOLUTIONHOOD is used as a maneuver to position the claim and its supporting argument. As structured, the locus of intended effect (segment 2), follows logically from the argument:

```
solutionhood(
  1,
  evidence(
    evidence(
      cause(
        circumstance(
          5,6),
        cause(
          9,
          means(
            7,8))),
      cause(
     3,4)),
      2))
```

Carlson and Marcu (2001) do not define a SOLUTIONHOOD relation *per se*, but rather specify an array of more finely-grained solution-oriented relations. Of particular concern are PROBLEM-SOLUTION, QUESTION-ANSWER, and STATEMENT-RESPONSE. For each of these, there are three subtypes. This, according to Carlson and Marcu, is necessary because sometimes the problem will be more important than the solution, sometimes the solution will be more important than the problem, and sometimes they will be of equal importance. Similar subtypes are specified for CONSEQUENCE, EVALUATION, and INTERPRETATION. That there should be such subtypes is not without precedent. Mann and Thompson used similar pairings for causal relations (VOLITIONAL-CAUSE, NON-VOLITIONAL CAUSE, and VOLITIONAL-RESULT, NON-VOLITIONAL RESULT). More recently, Stede et al. (2017) specified satellite and nuclear subtypes for the EVALUATION relation.

In SOLUTIONHOOD, the essence of the relation is that one part provides a solution to a problem presented by the second part (Mann & Thompson, 1986). The reader recognizes that the nucleus is a solution to the problem presented in the satellite. Nothing is stipulated as to the importance of one part over the other. In Carlson and Marcu’s PROBLEM-SOLUTION, one part presents a problem, the other presents a solution. They do stipulate that one part might be more important than the other, and that this will determine the relation. *Importance* here has to do with how salient or essential each part is. So if the problem is deemed more important, it is coded as the nucleus, and if the solution is more important, then it is coded as the nucleus. This exemplifies a fundamental difference between Mann and Thompson’s vision for what RST is and Carlson and Marcu’s approach. For Mann and Thompson the nuclearity of a relation is determined by specific constraints on the spans. For SOLUTIONHOOD this means that one part must present a problem and the other must be a solution to the problem. From this the satellite and nucleus are determined. Any determination of importance, salience, or essentiality follows as a consequence of conformance to the defined constraints. For Carlson and Marcu, the nuclearity of a relation is determined by a identification of the relative importance, salience, or essentiality of the spans. This is amounts to saying that the nuclearity of the spans is determined by the nuclearity of the spans. The difficulty presented by this circular reasoning is not merely hypothetical. Determining relational salience on the basis of something other than functional constraints adds a subjective feature to the analysis. In their example of PROBLEM-SOLUTION-S, because the problem is deemed more important than the solution, Carlson and Marcu assigned the role of nucleus to the problem and satellite to the solution. Here is the text:

1) *Despite the drop in prices for thoroughbreds, owning one still isn't cheap. At the low end, investors can spend $15,000 or more to own a racehorse in partnership with others. At a yearling sale, a buyer can go solo and get a horse for a few thousand dollars. But that means paying the horse's maintenance; on average, it costs $25,000 a year to raise a horse.*
2) *For those looking for something between a minority stake and total ownership, the owners' group is considering a special sale where established horse breeders would sell a 50% stake in horses to newcomers.*

Clearly, the first span is the problem and the second span is the solution. So by definition, the relation is SOLUTIONHOOD, or some sort of PROBLEM-SOLUTION. But it is unclear how this perception, that the satellite as more salient than the nucleus arises. On the contrary, the context in which this text is drawn suggests otherwise. The second span is the concluding paragraph from a *Wall Street Journal* article entitled *Racehorse Breeders Bet the Average Joe Would Pay for a Piece of Thoroughbred*. In my view, allowing the relational intentionality to determine the salience, and hence the relation, rather than an arbitrary designation of salience determining the relation, is the preferred approach for discovering rhetorical structure.

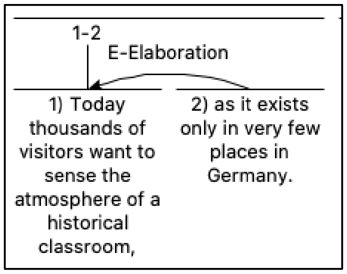

1-2
E-Elaboration
1) Today thousands of visitors want to sense the atmosphere of a historical classroom,
2) as it exists only in very few places in Germany.

Figure 4: Example of E-ELABORATION, as defined by (Stede, Taboada, & Das, 2017).

While subtyping SOLUTIONHOOD in terms of salience seems questionable, even if accepted, the inferentiality of its relational propositions remains consistent. Within the Boolean domain of problem-solution matchups, the solution follows from the problem, not the other way on. Although it makes sense to say

I'm hungry.
THEREFORE, let's go to the Fuji Gardens

the coherence of

The owners' group is considering a special sale.
THEREFORE, owning one still isn't cheap.

is questionable at best, except perhaps from a cynical perspective unsupported by the text. If the writer's intent is that the solution is less than adequate, its relation to the problem may not be SOLUTIONHOOD at all, but possibly ELABORATION or even a causal relation. The bottom line then is that if the relation is SOLUTIONHOOD or any of its variants, then inferentiality flows from problem to solution, irrespective of nuclearity.

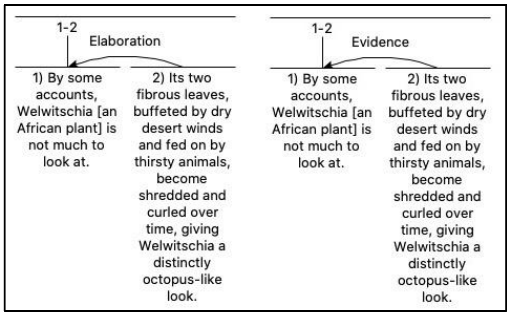

1-2
Elaboration
1) By some accounts, Welwitschia [an African plant] is not much to look at.
2) Its two fibrous leaves, buffeted by dry desert winds and fed on by thirsty animals, become shredded and curled over time, giving Welwitschia a distinctly octopus-like look.
1-2
Evidence
1) By some accounts, Welwitschia [an African plant] is not much to look at.
2) Its two fibrous leaves, buffeted by dry desert winds and fed on by thirsty animals, become shredded and curled over time, giving Welwitschia a distinctly octopus-like look.

Figure 5: ELABORATION and/or EVIDENCE

6 Inferential ELABORATION

ELABORATION is among the most frequently used relations (Cardoso, Taboada, & Pardo, 2013; Carlson & Marcu, 2001). It has also been the subject of controversy. Knott, Oberlander, O'Donnell, and Mellish (2001) have sometimes been cited as advocating that ELABORATION should be removed from the RST relations set (e.g., Louis & Nenkova, 2010; Marcu & Echihabi, 2002; Prévot, Vieu, & Asher, 2009; Taboada & Mann, 2006); however, their concern was more limited than that. Their objection was to one particular ELABORATION subtype: *Object-Attribute*. They considered this relation to be idiosyncratic. The distinctive characteristic of this subtype is that, from their perspective, it is not really a relation among discourse units, but rather a relation between a clause and an element within another clause. Stede et al. (2017) addressed this objection at least in part by defining two types of elaboration. In the first type, called ELABORATION, the satellite provides details or more information on the state of a affairs described in the nucleus. The second type, called E-ELABORATION, the satellite may refer not to the situation presented in the nucleus but to some element or entity mentioned in the nucleus (Stede et al., 2017), as illustrated in Figure 4. This appears consistent with the definition provided by Mann and Thompson, with the exception that the Object-Attribute subtype is set apart as its own relation. Carlson and Marcu treat ELABORATION as a general class, rather than a relation, subclassing it into eight separate relations. Although these are sufficiently distinct to be useful by an analyst or parser, from a rhetorical standpoint, they all accomplish the same thing – the assertion of the satellite is intended to increase the likelihood the reader will understand the nucleus. In this respect these relations are similar to BACKGROUND, and have the same logical generalization—similar, but not identical. One obvious difference is that the BACKGROUND satellite usually precedes the nucleus, and thus anticipates the need for supportive information (in this way BACKGROUND is similar to PREPARATION, except its domain is *comprehension* rather than *interest*). A more fundamental difference is that BACKGROUND is more general than ELABORATION.

Considered rhetorically we tend to reach one type of definition, based on writer intent, but considered solely from the perspective semantics, we arrive at something else. If in ELABORATION the satellite presents additional detail about the situation or some element presented in the nucleus, the question remains as to what the writer might hope to accomplish when employing the relation. We cannot look to Mann and Thompson, Stede, or Carlson for guidance there.

Unfortunately, the definition provided in one of my earlier papers (Potter, 2019) is also less than helpful. There, the relation is said to include an inference between the nucleus and the satellite, $(n \rightarrow s)$. The inference of s from n was due to the specification that the satellite be inferentially accessible in the nucleus, as originally defined by Mann and Thompson (1988). And since ELABORATION is a subject matter relation, neither n nor s are controversial, so both were treated as asserted, such that the definition given is $(s \wedge n \wedge (n \rightarrow s))$. So there are a couple of problems here. First, regarding the inferential accessibility, this merely establishes *relevance*, not *intended effect*, and second, even the definition were discursively representative, it would be logically redundant, since anytime $(s \wedge n)$, it will follow that $(n \rightarrow s)$. But more to the point, this definition is *not* discursively representative.

As Hobbs (1979) observed, an elaboration enhances the reader's understanding by providing additional information. Thus for RST, the Boolean domain of ELABORATION should be seen as one of clarification, and the inferential path is from satellite to nucleus, not nucleus to satellite. This is the case for all forms of ELABORATION. One way or another, the satellite supports the nucleus, making it more informative. That is its Boolean domain. Its logic is therefore *modus ponens*, $(((s \rightarrow n) \wedge s) \rightarrow n)$. It is not always easy to determine whether an elaboration should be read semantically or pragmatically. The example shown in Figure 5, concerning the appearance of the African plant, *Welwitschia*, can be analyzed as *elaboration(2,1)*, and this also may help resolve any doubts the reader might have as to the plant's ugliness, hence *evidence(2,1)*. Either way, the logic is the same.

7 Conclusion

The possibility that relational propositions might support an alignment of discourse with propositional logic appears to have occurred to Mann and Thompson in their development of RST. While refraining from commitment to this conceptualization, they hint at its possibility in their early publications on relational propositions (Mann & Thompson, 1983, 1986). However, the logic of relational propositions maps readily from their original vision. The abstraction of RST analyses as logical expressions provides a means for mapping argumentative inference with high granularity, and with traceability back to the text. A key enabler for this process is the alignment of Boolean domains with writer intentionality. While this multiplicity of Boolean domains is, so far as I know, a novel concept for argument mining, applications of Boolean logic beyond truth functional domains are by no means new, having an extensive history in circuit design, set theory, digital logic, and database query languages.

The inspirational notion here is of an interactive inference mining browser that would perform automated RST analysis of free texts, restate the RST analysis as a nested relational proposition, and generate a logical expression representing the inferential processes in the text. This could be integrated with other tools for identification of argumentative structures. There remain fundamental issues to be addressed. Problems with relation definition, such as those examined in this paper need to be resolved. As things stand, there are several *de facto* standards for RST analysis, none of them fully adequate and yet all seemingly frozen in time. Tall monuments cast long shadows. Hopefully what I have presented here will be useful, and if not in fully solving any problems then in at least in taking steps toward their solution.

References

Eric Abelen, Gisela Redeker, & Sandra Thompson. 1993. The rhetorical structure of US-American and Dutch fund-raising letters. *Text - Interdisciplinary Journal for the Study of Discourse, 3*, 323-350.

Nicholas Asher, & Alex Lascarides. 2003. *Logics of conversation*. Cambridge, UK: Cambridge University Press.

Moshe Azar. 1995. Argumentative texts in newspapers. In F. H. van Eemeren, R. Grootendorst, J. A. Blair, & C. H. Willard (Eds.), *Proceedings of the Third ISSU Conference on Argumentation* (Vol. 3, pp. 493–500). Amsterdam: University of Amsterdam.

Moshe Azar. 1997. Concession relations as argumentation. *Text, 17*(3), 301-316.

Moshe Azar. 1999. Argumentative text as rhetorical structure: An application of rhetorical structure theory. *Argumentation, 13*(1), 97-114.

Paula C.F. Cardoso, Maite Taboada, & Thiago A.S. Pardo. 2013. On the contribution of discourse structure to topic segmentation. In *Proceedings of the SIGDIAL 2013 Conference* (pp. 92-96). Metz, France: Association for Computational Linguistics.

Lynn Carlson, & Daniel Marcu. 2001. *Discourse tagging reference manual* (TR-2001-545). Retrieved from Marina del Rey, CA: ftp://ftp.isi.edu/isi-pubs/tr-545.pdf

Laurence Danlos. 2008. Strong generative capacity of RST, SDRT and discourse dependency DAGSs. In Anton Benz & Peter Kühnlein (Eds.), *Constraints in discourse* (pp. 69–95). Amsterdam: Benjamins.

N. Ye. Doronkina. 2017. Complex argumentation in the context of rhetorical structure theory in scientific discourse. *Journal of the National Technical University of Ukraine "KPI": Philology and Educational Studies, 9*, 22-26.

Boris Galitsky, Dmitry Ilvovsky, & Sergey O. Kuznetsov. 2018. Detecting logical argumentation in text via communicative discourse tree. *Journal of Experimental & Theoretical Artificial Intelligence, 30*(5), 637-663. doi:10.1080/0952813X.2018.1467492

Maria Garcia-Villalba, & Patrick Saint-Dizier. 2012. Some facets of argument mining for opinion analysis. In Bart Verheij, Stefan Szeider, & Stefan Woltran (Eds.), *Proceeding of the 2012 conference on Computational Models of Argument (COMMA 2012)*. Amsterdam: IOS Press.

Montserrat González, & Montserrat Ribas. 2008. The construction of epistemic space via causal connectives. In Istvan Kecskes & Jacob Mey (Eds.), *Intention, common ground and the egocentric speaker-hearer* (pp. 127-149). Berlin: de Gruyter.

Nancy L. Green. 2010. Representation of argumentation in text with rhetorical structure theory. *Argumentation, 24*(2), 181-196. doi:DOI: 10.1007/s10503-009-9169-4

Jeroen Groenendijk. 2009. Inquisitive semantics: Two possibilities for disjunction. In Peter Bosch, David Gabelaia, & Jérôme Lang (Eds.), *Logic, language, and computation* (pp. 80-94). Berlin, Heidelberg: Springer Berlin Heidelberg.

Jaakko Hintikka. 2007. *Socratic epistemology: Explorations of knowledge-seeking by questioning*. New York: Cambridge University Press.

Jerry R. Hobbs. 1979. Coherence and coreference. *Cognitive Science, 3*, 67-90.

Jerry R. Hobbs. 1985. *On the coherence and structure of discourse* (CSLI-85-37). Retrieved from Stanford, CA: http://www.isi.edu/~hobbs/ocsd.pdf

Oier Imaz, & Mikel Iruskieta. 2017. Deliberation as genre: Mapping argumentation through relational discourse structure. In *Proceedings of the 6th Workshop Recent Advances in RST and Related Formalisms* (pp. 1-10). Santiago de Compostela, Spain: Association for Computational Linguistics.

Alistair Knott, Jon Oberlander, Michael O'Donnell, & Chris Mellish. 2001. Beyond elaboration: The interaction of relations and focus in coherent text. In Ted Sanders, Joost Schilperoord, & Wilbert Spooren (Eds.), *Text Representation: Linguistic and Psycholinguistic Aspects* (pp. 181-196). Amsterdam: John Benjamins.

Annie Louis, & Ani Nenkova. 2010. Creating local coherence: An empirical assessment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL* (pp. 313-316). Los Angeles, California: Association for Computational Linguistics.

William C. Mann, & Sandra A. Thompson. 1983. *Relational propositions in discourse*. Marina del Rey, CA: Information Sciences Institute.

William C. Mann, & Sandra A. Thompson. 1986. Relational propositions in discourse. *Discourse Processes, 9*(1), 57-90.

William C. Mann, & Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse, 8*(3), 243-281.

William C. Mann, & Sandra A. Thompson. 2000. Two views of rhetorical structure theory. In *Proceedings of the 10th Annual Meeting of the Society for Text and Discourse*. Lyon, France: Society for Text and Discourse.

Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*. Cambridge, MA: MIT Press.

Daniel Marcu, & Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 368-375). Philadelphia.

Jelena Mitrović, Cliff O'Reilly, Miljana Mladenović, & Siegfried Handschuh. 2017. Ontological representations of rhetorical figures for argument mining. *Argument & Computation, 8*(3), 267-287.

Elena Musi, Debanjan Ghosh, & Smaranda Muresan. 2018. ChangeMyView through concessions: Do concessions increase persuasion? *Dialogue and Discourse, 9*(1), 107–127.

Nick Nicholas. 1994. *Problems in the application of rhetorical structure theory to text generation.* (masters thesis), University of Melbourne, Melbourne, Australia.

Andreas Peldszus. 2016. Rhetorical structure and argumentation structure in monologue text. In *Proceedings of the 3rd Workshop on Argument Mining* (pp. 103-112). Berlin, Germany: Association for Computational Linguistics.

Andreas Peldszus, & Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), 7*(1), 1-31.

Andrew Potter. 2007. A discourse approach to explanation aware knowledge representation. In Thomas Roth-Berghofer, Stefan Schulz, David B. Leake, & Daniel Bahls (Eds.), *Explanation-aware computing: Papers from the 2007 AAAI Workshop* (pp. 56-63). Menlo Park, CA: AAAI Press.

Andrew Potter. 2008a. Generating discourse-based explanations. *Künstliche Intelligenz, 22*(2), 28-31.

Andrew Potter. 2008b. Linked and convergent structures in discourse-based reasoning. In Thomas Roth-Berghofer, Stefan Schulz, Daniel Bahls, & David B. Leake (Eds.), *Proceedings of the 3rd International Explanation Aware Computing Workshop (ExaCt 2008)* (pp. 72-83). Patras, Greece.

Andrew Potter. 2009. Discourse-based reasoning for controlled natural languages. In Norbert E. Fuchs (Ed.), *Proceedings of the Workshop on Controlled Natural Language (CNL 2009).* Marettimo Island, Italy.

Andrew Potter. 2010. Rhetorical compositions for controlled natural languages. In Norbert E. Fuchs (Ed.), *Controlled Natural Language: Workshop on Controlled Natural Language, CNL 2009, Marettimo Island, Italy, June 8-10, 2009, Revised Papers* (pp. 21-35). Heidelberg: Springer.

Andrew Potter. 2019. Reasoning between the lines: A logic of relational propositions. *Dialogue and Discourse, 9*(2), 80-110.

Andrew Potter. 2020. The rhetorical structure of Modus Tollens: An exploration in logic-mining. In Allyson Ettinger, Ellie Pavlich, & Brandon Prickett (Eds.), *Proceedings of the Society for Computation in Linguistics* (Vol. 3, pp. 170-179). New Orleans, LA: SCiL.

Andrew Potter. 2021. Text as tautology: an exploration in inference, transitivity, and logical compression. *Text & Talk*.

Laurent Prévot, Laure Vieu, & Nicholas Asher. 2009. Une formalisation plus précise pour une annotation moins confuse: la relation d'Élaboration d'entité. *French Language Studies, 19*(207-228).

Andrea Rocci. 2021. Diagramming counterarguments: At the interface between discourse structure and argumentation structure. In Ronny Boogaart, Henrike Jansen, & Maarten van Leeuwen (Eds.), *The Language of Argumentation* (pp. 143-166). Cham: Springer International Publishing.

Gilbert Ryle. 1949. *The Concept of Mind*. Chicago, IL: University of Chicago Press.

Ted J M Sanders, W P M Spooren, & L G M Noordman. 1992. Toward a taxonomy of coherence relations. *Discourse Processes, 15*, 1-35.

Manfred Stede. 2008. RST revisited: Disentangling nuclearity. In Cathrine Fabricius-Hansen & Wiebke Ramm (Eds.), *'Subordination' versus 'coordination' in sentence and text – from a cross-linguistic perspective* (pp. 33-58). Amsterdam: Benjamins.

Manfred Stede. 2020. From coherence relations to application: The case of contrast and argumentation. In *Computational Linguistics and Intellectual Technologies*.

Manfred Stede, Maite Taboada, & Debopam Das. 2017. *Annotation guidelines for rhetorical structure*. Retrieved from Potsdam and Burnaby: http://www.sfu.ca/~mtaboada/docs/research/RST_Annotation_Guidelines.pdf

Maite Taboada, & William C. Mann. 2006. Rhetorical structure theory: Looking back and moving ahead. *Discourse Studies, 8*(3), 423-459.

Sandra A. Thompson, & William C. Mann. 1986. A discourse view of Concession in written English. In *Proceedings of the Second Annual Meeting of the Pacific Linguistics Conference* (pp. 435-447). Eugene, Oregon: University of Oregon, Eugene.

Shujun Wan, Tino Kutschbach, Anke Lüdeling, & Manfred Stede. 2019. RST-Tace A tool for automatic comparison and evaluation of RST trees. In Amir Zeldes, Debopam Das, Erick Maziero Galani, Juliano Desiderato Antonio, & Mikel Iruskieta (Eds.), *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019* (pp. 88-96). Minneapolis, Minnesota: Association for Computational Linguistics.

Wing-Kwong C. Wong. 1986. *A theory of argument coherence* (TR86-29). Retrieved from Austin, Texas:

Adam Wyner, & Jodi Schneider. (2012). *Arguing from a point of view*. Paper presented at the First International Conference on Agreement Technologies, Dubrovnik, Croatia.

Amir Zeldes, & Yang Liu. 2020. A neural approach to discourse relation signal detection. *Dialogue and Discourse, 11*(2), 1-33.

Learning Argument Structures with Recurrent Neural Network Grammars

Ryo Yoshida and **Yohei Oseki**
The University of Tokyo
{yoshiryo0617, oseki}@g.ecc.u-tokyo.ac.jp

Abstract

In targeted syntactic evaluations, the syntactic competence of language models (LMs) has been investigated through various syntactic phenomena, among which one of the important domains has been *argument structure*. Argument structures in head-initial languages have been exclusively tested in the previous literature, but may be readily predicted from lexical information of verbs, potentially overestimating the syntactic competence of LMs. In this paper, we explore whether argument structures can be learned by LMs in head-final languages, which could be more challenging given that argument structures must be predicted before encountering verbs during incremental sentence processing, so that the relative weight of syntactic information should be heavier than lexical information. Specifically, we examined double accusative constraint and double dative constraint in Japanese with the sequential and hierarchical LMs: *n*-gram model, LSTM, GPT-2, and Recurrent Neural Network Grammar (RNNG). Our results demonstrated that the double accusative constraint is captured by all LMs, whereas the double dative constraint is successfully explained only by the hierarchical model. In addition, we probed incremental sentence processing by LMs through the lens of surprisal, and suggested that the hierarchical model may capture deep semantic roles that verbs assign to arguments, while the sequential models seem to be influenced by surface case alignments. We conclude that the explicit hierarchical bias is essential for LMs to learn argument structures like humans.

1 Introduction

Recently, artificial neural networks have had a great impact on the field of Natural Language Processing. Nevertheless, despite the improvement brought by the neural network, it is an open question what linguistic knowledge neural language models (LMs) can learn from the next word prediction task. One line of research peeking into the neural network "black box" is the targeted syntax evaluations with controlled sentences designed to reveal whether the LMs have learned specific syntactic knowledge consistent with human acceptability judgements. (e.g., Lau et al., 2017). Using this method, previous work has shown that these models successfully learn a variety of syntactic knowledge such as subject-verb number agreement (Linzen et al., 2016; Marvin and Linzen, 2018; Wilcox et al., 2018).

In targeted syntax evaluations, one of the important domains has been *argument structure*. Previous work suggested that neural LMs have the ability to capture argument structures (Kann et al., 2019; Warstadt et al., 2020), but in head-initial languages exclusively tested in the previous literature, argument structures may be predicted from lexical information of verbs, potentially overestimating the syntactic competence of the LMs. In addition, although targeted syntax evaluation to test other linguistic knowledge has confirmed the advantage of syntactic bias (Kuncoro et al., 2018; Wilcox et al., 2019; Futrell et al., 2019), hierarchical models such as Recurrent Neural Network Grammars (RNNGs, Dyer et al., 2016) have not been evaluated for verb argument structures.

In this paper, we will examine the effect of syntactic bias on learning verb argument structures, using more challenging head-final language, Japanese. In Japanese, argument structures must be predicted before encountering verbs during incremental sentence processing, such that the relative weight of syntactic information should be heavier than lexical information. We specifically focus on the double accusative constraint (e.g., Harada, 1975, 1986; Shibatani, 1978; Hiraiwa, 2002, 2010) and the double dative constraint in Japanese. The double accusative constraint prohibits the occur-

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 101-111.
Held on-line February 7-9, 2022

Previous literature	English	Italian	Russian	French	German	Hebrew	Basque	Japanese
Linzen et al. (2016), Marvin and Linzen (2018), Jumelet and Hupkes (2018), Chowdhury and Zamparelli (2018, 2019), Wilcox et al. (2018, 2019), Futrell et al. (2019), Warstadt et al. (2019a,b, 2020), Chaves (2020), Da Costa and Chaves (2020), Hu et al. (2020)	✓							
Gulordava et al. (2018)	✓	✓	✓			✓		
Ravfogel et al. (2018)							✓	
An et al. (2019)	✓			✓				
Mueller et al. (2020)	✓		✓	✓	✓	✓		

Table 1: Summary of the previous literature on targeted syntactic evaluations. Works for English are shown above the horizontal line and works for other European languages are shown below the horizontal line.

rences of two or more NPs marked with the accusative case particle *o* within the same clause, and the double dative constraint is the restriction on the case taken by verbs. We will test these constraints with the sequential and hierarchical LMs, *n*-gram, LSTM, GPT-2 (Radford et al., 2019) and Recurrent Neural Network Grammars (RNNGs). As a result, we demonstrated that the double accusative constraint could be captured by all LMs, whereas the double dative constraint is successfully explained only by the hierarchical model. In addition, we analyzed the phrase-by-phrase surprisal of the LMs, and suggested that the hierarchical model may capture deep semantic roles that verbs assign to arguments, while the sequential models are influenced by surface case alignments. This result suggests that the double accusative constraint, which is a constraint to spell out the surface case, can be solved well by the sequential model, but the double dative constraint, which is a constraint at the level of the deep semantic role that verbs assign to arguments, can be solved well only by the hierarchical model. Taken together, we conclude that the explicit hierarchical bias is essential for LMs to learn the human-like syntactic competence to process argument structures.

Another important contribution of this paper is that, to the best of our knowledge, it was the first attempt to conduct targeted syntax evaluation using Japanese. The goal of natural language processing community is to build a LM having language independent general language processing ability, but so far targeted syntax evaluation has been done mainly for English (above the horizontal line in Table 1) and other European languages (below the horizontal line in Table 1). In order to achieve the goal, it is important to evaluate the syntactic competence of LMs for non-European languages.

2 Methods

To investigate the effect of explicitly modeling hierarchical structures, we train linear LMs and a hierarchical LM. In order to eliminate the effect of the amount of training data, we trained all LMs on the same training data. In addition, we restricted our evaluation to left-to-right LMs corresponding to incremental sentence processing, to make LMs predict the verb argument structure before they see the verb. We used the same model sizes reported in the papers proposing each model (Table 2).

2.1 Language Models

Long Short-Term Memory (LSTM): LSTMs are a sequential model using the recurrent neural network architecture (Hochreiter and Schmidhuber, 1997). We used a 2-layer LSTM with 256 hidden and input dimensions. The implementation by Gulordava et al. (2018) was employed.[1]

GPT-2: GPT-2 is a sequential model using the Transformer architecture (Vaswani et al., 2017). We used the same architecture of GPT-2 small (Radford et al., 2019) with 12 layers and 756 hidden and input dimensions. The implementation by Huggingface's Transformer package (Wolf et al., 2020) was employed.

Recurrent Neural Network Grammar (RNNG): RNNGs are a hierarchical model which explicitly models hierarchical structures (Dyer et al., 2016).

[1] https://github.com/facebookresearch/colorlessgreenRNNs

Language Model	#Layers	#Hidden dimensions	#Input dimensions
LSTM	2	256	256
GPT-2	12	768	768
RNNG	2	256	256

Table 2: Model sizes of neural LMs evaluated in this paper.

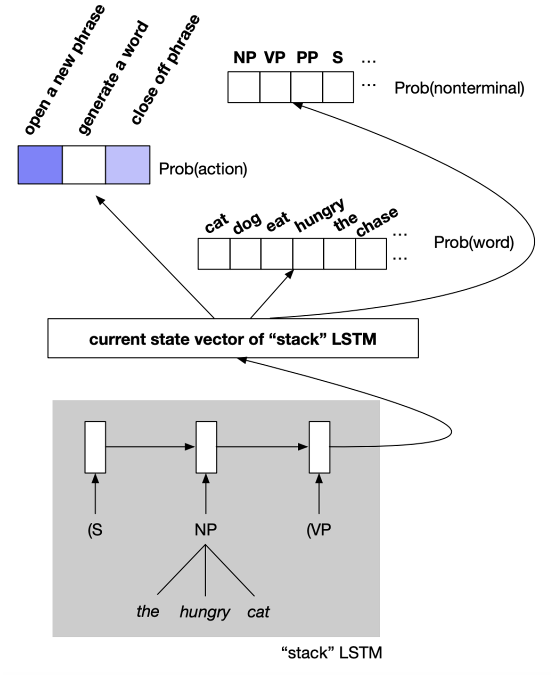


Figure 1: The architecture of RNNGs used in this paper. This figure is reproduced from Hale et al. (2018).

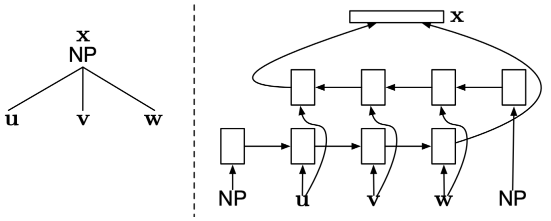


Figure 2: "Syntactic composition" function that is executed during a REDUCE action. This figure is reproduced from Dyer et al. (2016).

In this paper, we used stack-only RNNGs (Kuncoro et al., 2017). RNNGs generate trees such as "(S (NP *The hungry cat*) (VP *meows*))"; each of the elements is encoded as a vector and stored in a stack, which is illustrated inside the gray box in Figure 1. At each step of generation, one of the following three actions is selected based on the current state of the stack, which is encoded as a vector by stack LSTM:

- **NT(X)** introduces a nonterminal X that is encoded as a vector onto the top of the stack. This action generates an open nonterminal "(X".
- **GEN(x)** introduces a terminal symbol x that is encoded as a vector onto the top of the stack. This action generates a terminal symbol "x".
- **REDUCE** triggers "syntactic composition" function, which creates a new single vector that represents a phrase X from the elements of its children in the stack. For example, "(NP *The hungry cat*)" is represented by a new single vector by this action.

If NT(X) or GEN(x) is selected, which open nonterminal or word is generated is selected based on the same vector that represents the current state of the stack.

If REDUCE is selected, "syntactic composition" function is executed by bidirectional LSTM (Figure 2). In both directions, a nonterminal vector such as "(NP" is input first, and then its children vectors such as "**u**", "**v**" and "**w**" are input in forward or reverse order. After all the children vectors are input, the phrase vector "**x**" is calculated from the output of the forward and reverse LSTMs.

We used RNNGs that had a 2-layer stack LSTM with 256 hidden and input dimensions. The implementation by Noji and Oseki (2021) was employed.[2] RNNGs were given the correct tree structures only during training, so we used word-synchronous beam search (Stern et al., 2017) to inference tree structures behind terminal subwords during evaluation. We set the action beam size to 100, the word beam size to 10, and the fast track to 1.

***n*-gram:** As a baseline, we also train 5-gram LM using KenLM.[3]

[2] https://github.com/aistairc/rnng-pytorch

[3] https://github.com/kpu/kenlm

2.2 Training data

All LMs were trained on the National Institute for Japanese Language and Linguistics Parsed Corpus of Modern Japanese (NPCMJ), that comprises 67,018 sentences annotated with tree structures.[4] The sentences were split into subwords by a byte-pair encoding (Sennrich et al., 2016).[5] LSTM, GPT-2, and *n*-gram used only terminal subwords, while RNNGs used terminal subwords and tree structures. All neural LMs (LSTM, GPT-2, and RNNGs) were given one sentence at a time and were trained for 40 epochs and 3 times with different random seeds.[6]

2.3 Acceptability judgements with LMs

Recently, many efforts have been made on the evaluation of the syntactic competence of LMs. Previous work (e.g., Linzen et al., 2016; Marvin and Linzen, 2018; Wilcox et al., 2018) evaluated whether LMs assign a higher probability to an acceptable sentence than to an unacceptable one, using a minimal pair such as (1).

(1) a. The hungry cat meows.

b. *The hungry cat meow.

There are several methods to measure an LM's preference between two sentences in a minimal pair. One of them is prediction task, which compares a probability of grammatically critical position. For example, in the example in (1), we would expect the model to predict $p(\text{meows}|\text{The hungry cat}) > p(\text{meow}|\text{The hungry cat})$. However, the prediction task setting is not applicable when grammaticality is determined by the interaction of several words or when the information necessary to determine grammaticality does not appear in the left context. In this paper, we use the more general full-sentence setting (Marvin and Linzen, 2018; Warstadt et al., 2020), which compares a probability of the two complete sentences. For example, in the example in (1), we would expect the model to predict $p(\text{The hungry cat meows}) > p(\text{The hungry cat meow})$.

3 Targeted argument structures

3.1 Double accusative constraint

Japanese has a constraint that prohibits the occurrences of two or more NPs marked with the accusative case particle *o* in the same clause. This constraint is called "double accusative constraint", and have attracted considerable interest in the study of Japanese syntax (e.g., Harada, 1975, 1986; Shibatani, 1978; Hiraiwa, 2002, 2010). One example of double accusative constraint is given in (2):

(2) a. Ken-ga Naomi-**ni/o** gakko-ni
Ken-Nom Naomi-Dat/Acc school-Dat
ik-ase-ta
go-Caus-Past
'Ken made Naomi go to school.'

b. Ken-ga Naomi-**ni** sono-hon-**o**
Ken-Nom Naomi-Dat Dem-book-Acc
yom-ase-ta
read-Caus-Past
'Ken made Naomi read the book.'

c. *Ken-ga Naomi-**o** sono-hon-**o**
Ken-Nom Naomi-Acc Dem-book-Acc
yom-ase-ta
read-Caus-Past
'Ken made Naomi read the book.'

As shown in (2a), when the object NP is marked with the dative case particle *ni*, the causee NP can be marked with either the dative case particle *ni* or the accusative case particle *o*. However, as shown in (2bc), when the object NP is marked with the accusative case particle *o*, the causee NP cannot be marked with the accusative case particle *o* (2c), but must be marked with the dative case particle *ni* (2b).

We can assess the syntactic competence of LMs on double accusative constraint by examining whether LMs assign a higher probability to (2b) than (2c), where both arguments are marked with the accusative case particle *o* within the same sentence. For this purpose, 22 minimal pairs of the (2bc) pattern made by Tamaoka et al. (2018) were collected and the probabilities of the two sentences were compared for each minimal pair. We confirmed that case markers are tokenized into individual subword tokens.

3.2 Double dative constraint

Now we turn to another phenomenon on argument structures: double dative constraint. In Japanese,

[4] `http://npcmj.ninjal.ac.jp`

[5] Implemented in sentencepiece (Kudo and Richardson, 2018). We set character coverage to 0.9995, and vocabulary size to 8000

[6] Traces and semantic information were removed in the way described in Manning and Schutze (1999).

Language Model	Accuracy (%)
n-gram	72.7
LSTM	97.0 (± 2.1)
GPT-2	95.5 (± 3.7)
RNNG	**98.5** (± 2.1)

Table 3: The result of targeted syntactic evaluation on double accusative constraint. Average accuracies with standard deviations across different random seeds are reported.

Language Model	Accuracy (%)
n-gram	81.8
LSTM	89.4 (± 8.6)
GPT-2	86.4 (± 3.7)
RNNG	**100.0** (± 0.0)

Table 4: The result of targeted syntactic evaluation on double dative constraint. Average accuracies with standard deviations across different random seeds are reported.

case is marked with particles, and different verbs can take different case patterns. One example of double dative constraint is given in (3):

(3) a. Ken-ga Naomi-**o** gakko-**ni**
Ken-Nom Naomi-Acc school-Dat
oku-tta
take-Past
'Ken took Naomi to school.'

b. *Ken-ga Naomi-**ni** gakko-**ni**
Ken-Nom Naomi-Dat school-Dat
oku-tta
take-Past
'Ken took Naomi to school.'

As shown in (3a), double object verbs take three arguments: an NP marked with the nominative case particle *ga*, an NP marked with the accusative case particle *o*, and an NP marked with the dative case particle *ni*. Double object verbs cannot take an NP marked with the dative case instead of an NP marked with the accusative case (3b), resulting in unacceptable sentences.

We can assess the syntactic competence of LMs on double dative constraint by examining whether LMs assign a higher probability to (3a) than (3b). In order to make the results comparable to the double accusative constraint in the previous section, we contrast (3a) with (3b), where both arguments are marked with the dative case particle *ni* within the same sentence. For this purpose, 22 minimal pairs of the (3ab) pattern made by Tamaoka et al. (2018) were collected and the probabilities of the two sentences were compared for each minimal pair. We confirmed that case markers are tokenized into individual subword tokens.

4 Results

4.1 Double accusative constraint

The result of targeted syntactic evaluation on double accusative constraint is shown in Table 3. Average accuracies with standard deviations across different random seeds are reported. First, the baseline *n*-gram model underperformed the neural LMs. This result demonstrates that the dataset used to test the double accusative constraint cannot merely be solved with local information.

Second, among the neural LMs, the hierarchical model (RNNG) achieved the highest accuracy, while the sequential models (LSTM and GPT-2) also reached the near perfect performance. This result provide evidence supporting that the neural LMs can capture the double accusative constraint without explicitly modeling hierarchical structures.

4.2 Double dative constraint

The result of targeted syntactic evaluation on double dative constraint is shown in Table 4. Average accuracies with standard deviations across different random seeds are reported. First, the baseline *n*-gram model performed relatively well, but the performance is still lower than the neural LMs. This result indicates that the dataset used to test the double dative constraint can reasonably be solved with local information alone, but neural architectures may be required to reach the higher performance.

Second, among the neural LMs, the hierarchical model (RNNG) achieved the perfect accuracy, whereas the sequential models (LSTM and GPT-2) did not reach the near perfect performance with only slight improvements over the baseline *n*-gram model. This result provide evidence supporting that, unlike the double accusative constraint, the neural LMs can capture the double dative constraint only when explicitly modeling hierarchical structures.

5 Probing sentence processing

Sections 3.1 and 3.2 demonstrated that the explicit hierarchical bias may not be necessary for the double accusative constraint, but crucial for the double dative constraint. Why can the sequential models learn the double accusative constraint, but not the double dative constraint? In this section, following Futrell et al. (2019), we probe sentence processing and identify the phrases where LMs make different predictions for acceptable and unacceptable sentences by computing phrase-by-phrase surprisal of LMs. The following analyses include neural LMs to the exclusion of the *n*-gram model.

5.1 Methods

We probed sentence processing of LMs through the information-theoretic complexity metric called *surprisal* (Hale, 2001; Levy, 2008): $-\log p(\mathrm{segment}|\mathrm{context})$. In psycholinguistics, it is well known that humans predict next segments during incremental sentence processing, and the less predictable the segment is, the more surprising that segment is. The previous literature established that cognitive efforts measured from humans are proportional to surprisals computed from LMs (e.g., Smith and Levy, 2013; Frank and Bod, 2011; Frank et al., 2015). Building on this result, we probe sentence processing by LMs through the lens of surprisal.

5.2 Results

5.2.1 Double accusative constraint

Figure 3 shows phrase-by-phrase surprisal for the double accusative constraint. Phrasal surprisal was computed as the cumulative sum of surprisals of its constituent subwords. Average surprisals with standard errors across different items and random seeds are reported.

We observe that all LMs show the largest surprisal difference at the accusative case particle *o* marking the third NP. This observation suggests that the all LMs captured the double accusative constraint through consecutive case marking on the second and third NPs. Notice incidentally that only RNNG shows larger surprisal at the end of unacceptable sentences than acceptable sentences.

5.2.2 Double dative constraint

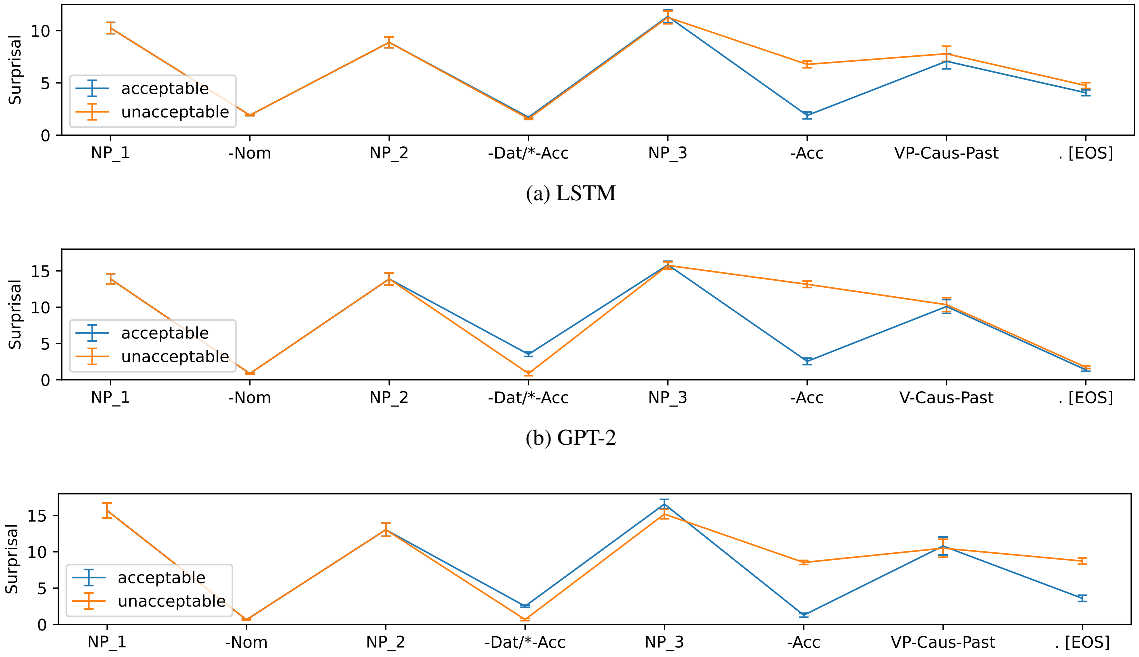
Figure 4 shows phrase-by-phrase surprisal for the double dative constraint. Phrasal surprisal was computed as the cumulative sum of surprisals of its constituent subwords. Average surprisals with standartd errors across different items and random seeds are reported.

First, unlike the double accusative constraint, we cannot observe the phrases where all LMs consistently show a large surprisal difference. Second, LSTM and GPT-2 show the largest surprisal difference at the dative case particle *o* marking the third NP, while RNNG shows the largest surprisal difference at the case particle marking the second NP. These observations suggest that the sequential models are more surprised when the dative case particle *ni* marks two NPs consecutively, while the hierarchical model is more surprised when the dative case particle *ni* marks the second NP incorrectly, which should be marked by the accusative case particle *o*.

In order to confirm this result, we statistically tested via paired-samples *t*-tests whether the "surprisal differences between acceptable and unacceptable sentences" are significantly different between the case particle marking the second NP (the phrase where the dative case particle marks one NP incorrectly) and the case particle marking the third NP (the phrase where the dative case particle marks two NPs consecutively). The result revealed that LSTM shows a significantly larger surprisal difference at the case particle marking the third NP ($p < 0.05$), while RNNG shows a significantly larger surprisal difference at the case particle marking the second NP ($p < 0.05$), but GPT-2 did not show any significant difference ($p = 0.067$). In other words, LSTM was more surprised when the dative case particle *ni* marks two NPs consecutively, while RNNGs were more surprised when the dative case particle *ni* marks the second NP incorrectly, which should be marked by the accusative case particle *o*, but GPT-2 was equally surprised at both phrases. The important conclusion here is that the hierarchical model (RNNG) not only achieved the perfect accuracy but also captured the double dative constraint for right reasons (i.e. incorrect case marking on the second NP), while the sequential models (LSTM and GPT-2) solved the double dative constraint for wrong reasons (i.e. consecutive case marking on the second and third NPs). In fact, as in (4), it is possible to have consecutive dative cases in Japanese, for example, when a NP marked by the dative case expresses time, and it is wrong to judge ungrammaticality on the basis of a series of dative cases.

(4)	Ken-ga	yoake-**ni**	gakko-**ni**	i-tta
	Ken-Nom	dawn-Dat	school-Dat	go-Past

'Ken went to school at dawn.'

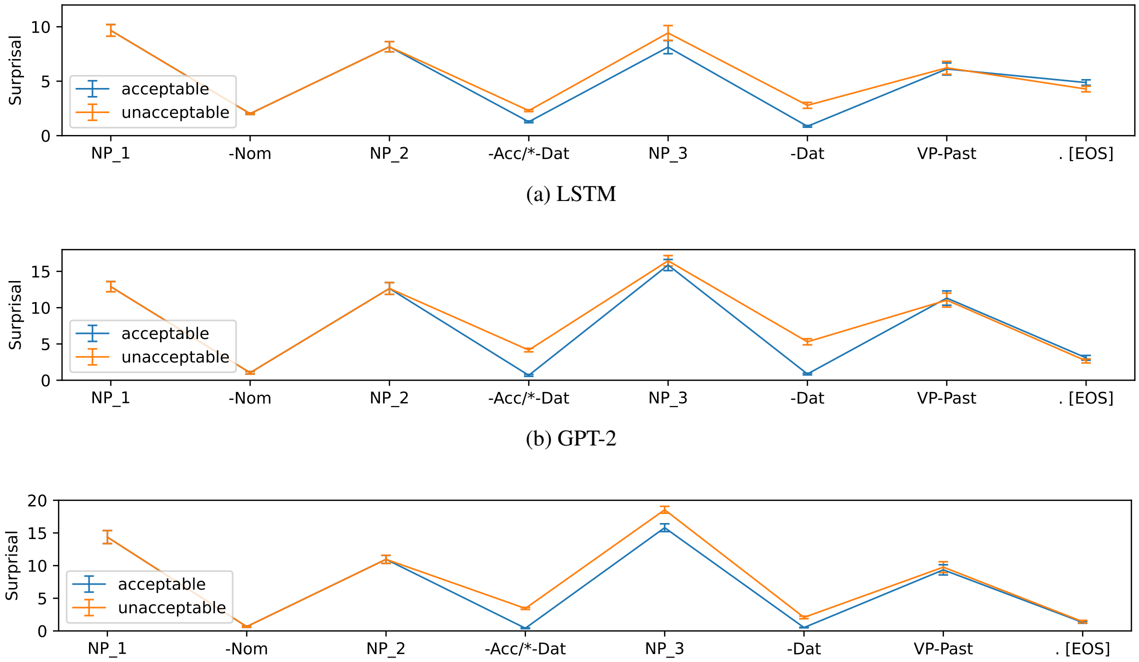


(a) LSTM

(b) GPT-2

(c) RNNG

Figure 3: Phrase-by-phrase surprisal for the double accusative constraint. Phrasal surprisal was computed as the cumulative sum of surprisals of its constituent subwords. Average surprisals with standard errors across different items and random seeds are reported.



(a) LSTM

(b) GPT-2

(c) RNNG

Figure 4: Phrase-by-phrase surprisal for the double dative constraint. Phrasal surprisal was computed as the cumulative sum of surprisals of its constituent subwords. Average surprisals with standard errors across different items and random seeds are reported.

6 General discussion

In summary, we demonstrated that all LMs can capture the double accusative constraint, while only the hierarchical model can solve the double dative constraint with the perfect accuracy. Moreover, further analyses of incremental sentence processing revealed that the double accusative constraint can be attributed to the phrase where the second and third NPs are marked consecutively, while the double dative constraint seems to be adequately captured by the hierarchical model at the phrase where the second NP is marked incorrectly. In this section, we discuss these results from the perspective of theoretical linguistics.

First, Hiraiwa (2010) proposes that the double accusative constraint is not a pure syntactic constraint, but an interface constraint on the spell-out of the accusative case; namely, the phonological constraint against realizing multiple occurrences of the accusative case value within the same domain. Interestingly, this proposal is consistent with our results in that the double accusative constraint is modeled by LMs through surface case alignments like consecutive case marking on the second and third NPs.

Second, the double dative constraint, on the other hand, seems to be a pure syntactic constraint, where NPs should be marked with the accusative case particle given deep semantic roles (i.e. theme) that verbs assign to arguments. Among the neural LMs tested above, only RNNG distinguished unacceptable sentences from acceptable sentences at the phrase where the second NP is marked incorrectly. Although GPT-2 also shows a similar trend to RNNG, the sequential models seem to be surprised for wrong reasons by consecutive case marking on the second and third NPs, which is not the critical point of the difference between acceptable and unacceptable sentences. This result may suggest that the sequential models cannot learn deep semantic roles that verbs assign to arguments and, alternatively, are strongly influenced by surface heuristics (McCoy et al., 2019). In contrast, the hierarchical model can learn those deep semantic roles by explicitly modeling hierarchical structures (Wilcox et al., 2020).

7 Limitations and future work

In this paper, we performed the targeted syntactic evaluation of LMs on argument structure in Japanese, which could be more challenging than English given that argument structures must be predicted before encountering verbs during incremental sentence processing. However, our results suggests that the dataset used in this paper may be too easy: even the baseline *n*-gram model can solve well (accuracy = 72.7% on double accusative constraint and 81.8% on double dative constraint). We should evaluate LMs on more challenging dataset to strengthen the argument in this paper.

In addition, in order to make the fair comparison of different architectures of the LMs, we trained all LMs on NPCMJ, the largest treebank in Japanese. However, since NPCMJ is relatively small (67,000 sentences), and the previous literature has shown that sequential models can reach the higher performance comparable to hierarchical models when trained on larger training data (Futrell et al., 2019), whether the results scale or not remains to be explored in future work.

Finally, this paper was the first attempt to conduct the targeted evaluation in Japanese, but only two syntactic phenomena on argument structures were examined in this paper. In order to scale the targeted syntactic evaluation, we plan to evaluate the syntactic competence of LMs on a wider range of syntactic phenomena in Japanese. We hope that this paper will motivate the targeted evaluation of the syntactic competence of LMs across languages.

8 Conclusion

In this paper, we explored whether argument structures can be learned by LMs in head-final languages, where argument structures must be predicted even before encountering following verbs during incremental sentence processing. Specifically, we examined double accusative constraint and double dative constraint in Japanese with the sequential and hierarchical LMs: *n*-gram model, LSTM, GPT-2, and RNNG. Our results demonstrated that the double accusative constraint could be captured by all LMs, whereas the double dative constraint is successfully explained only by the hierarchical model. In addition, we probed sentence processing by LMs through the lens of surprisal, and suggested that the hierarchical model may capture deep semantic roles that verbs assign to arguments, while the sequential models are influenced by surface case alignments. We conclude that the explicit hierarchical bias is essential for LMs to learn the human-like syntactic competence to process argument structures.

Acknowledgements

We would like to thank three anonymous reviewers of the *Society for Computation in Linguistics* for valuable comments and suggestions. This work was supported by JST PRESTO Grant Number JPMJPR21C2, and developed from a term paper submitted to the graduate course titled "Foundations of Linguistic Analysis I" offered at the Department of Language and Information Sciences, Graduate School of Arts and Sciences, University of Tokyo in Spring 2021.

References

Aixiu An, Peng Qian, Ethan Wilcox, and Roger Levy. 2019. Representation of constituents in neural language models: Coordination phrase as a case study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2888–2899, Hong Kong, China. Association for Computational Linguistics.

Rui Chaves. 2020. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 1–11, New York, New York. Association for Computational Linguistics.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shammur Absar Chowdhury and Roberto Zamparelli. 2019. An LSTM adaptation study of (un)grammaticality. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 204–212, Florence, Italy. Association for Computational Linguistics.

Jillian Da Costa and Rui Chaves. 2020. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 12–21, New York, New York. Association for Computational Linguistics.

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

Stefan Frank and Rens Bod. 2011. Insensitivity of the Human Sentence-Processing System to Hierarchical Structure. *Psychological science*, 22:829–34.

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.

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, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless Green Recurrent Networks Dream Hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

John Hale. 2001. A Probabilistic Earley Parser as a Psycholinguistic Model. In *Proceedings of the Second Conference of the North American Chapter of the Association for Computational Linguistics*, pages 159–166.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736, Melbourne, Australia. Association for Computational Linguistics.

S. I. Harada. 1986. "counter equi-np deletion". *Journal of Japanese Linguistics*, 11(1-2):157–202.

Shin-Ichi Harada. 1975. The functional uniqueness principle. *Attempts in linguistics and literature*, 2:17–24.

Ken Hiraiwa. 2002. Facets of case: On the nature of the double-o constraint. In *The proceedings of the 3rd Tokyo Psycholinguistics Conference (TCP 2002)*, pages 139–163. Citeseer.

Ken Hiraiwa. 2010. Spelling out the double-o constraint. *Natural Language & Linguistic Theory*, 28(3):723–770.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation*, 9(8):1735–80.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R. Bowman. 2019. Verb argument structure alternations in word and sentence embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 287–297.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. What Do Recurrent Neural Network Grammars Learn About Syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1249–1258, Valencia, Spain. Association for Computational Linguistics.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

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.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

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.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Hiroshi Noji and Yohei Oseki. 2021. Effective batching for recurrent neural network grammars. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4340–4352, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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*, pages 98–107, Brussels, Belgium. 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.

Masayoshi Shibatani. 1978. *Nihongo no bunseki*. Taishukan Publishing Company.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective Inference for Generative Neural Parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700, Copenhagen, Denmark. Association for Computational Linguistics.

Katsuo Tamaoka, Jingyi Zhang, and Toshiki Satoh. 2018. An experimental study on psychological reality of double accusative constraint by the maze task. *Studia Linguistica*, 32:115–130.

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 *Proceedings of NIPS*, pages 5998–6008.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019a. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019b. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. Structural supervision improves learning of non-local grammatical dependencies. 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 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Ryosuke Kohita, Roger Levy, and Miguel Ballesteros. 2020. Structural supervision improves few-shot learning and syntactic generalization in neural language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4640–4652, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Learning Stress Patterns with a Sequence-to-Sequence Neural Network

Brandon Prickett
University of Massachusetts Amherst
bprickett@umass.edu

Joe Pater
University of Massachusetts Amherst
pater@umass.edu

Abstract

We present the first application of modern neural networks to the well studied task of learning word stress systems. We tested our adaptation of a sequence-to-sequence network on the Tesar and Smolensky test set of 124 "languages", showing that it acquires generalizable representations of stress patterns in a very high proportion of runs. We also show that it learns restricted lexically conditioned patterns, known as stress windows. The ability of this model to acquire lexical idiosyncracies, which are very common in natural language systems, sets it apart from past, non-neural models tested on the Tesar and Smolensky data set.

1 Introduction

Some of the earliest work in computational phonology investigated the acquisition and representation of word stress patterns (Dresher and Kaye, 1990; Gupta and Touretzky, 1994). Stress is of interest because the extent of typological variation is relatively well understood, and because learning the patterns is non-trivial in various ways. A considerable amount of more recent work has focused on a data set created by Tesar and Smolensky (2000, henceforth *TS*); see further Jarosz (2013), Jarosz (2015) and Boersma and Pater (2016). The data set includes 124 languages that can be represented using 12 relatively standard Optimality Theoretic (Prince and Smolensky, 2004) constraints. Past work has tested various algorithms for weighting and ranking constraints to see which performed the best on this dataset (where performance was measured by how many of the 124 languages the models could learn with 100% accuracy).

In this paper, we explore how well a model without constraints, namely a sequence-to-sequence neural network, performs on the 124 languages. Two factors motivate this departure from constraint-based models: (i) a question of whether pre-specified structures like constraints[1] are necessary to represent and learn the stress patterns in the TS data set, and (ii) whether neural networks, which have the expressive power to capture both general and lexically specific patterns will be able to generalize stress patterns to novel data. We find that the sequence-to-sequence net does succeed in learning most of the languages, and that it generalizes to novel data, both when trained on the 124 TS languages and when trained on 6 novel patterns involving lexically conditioned stress. No previous research on the Dresher and Kaye parametric systems, or on the TS violable constraint systems, has provided a mechanism for the learning of lexically conditioned patterns – they can only acquire fully general ones. These results thus provide new challenges for future research using non-neural frameworks in this domain.

2 Background

The TS dataset was created to test an approach for handling *hidden structure* in phonology: how does a learner parse a form that it's being trained on when it hasn't learned all of the grammatical information needed for parsing in the first place? In the TS languages, this takes the form of stress patterns that are assumed to rely on foot-based structure to place the primary and secondary stress in a word. While the training data for a language includes mappings between underlying forms (strings of light and heavy syllables) to correctly stressed surface forms, that data does not include information about where the feet occur in the correct surface forms. An example of a piece of learning data in one of the TS languages is shown in (1), with *L* representing

[1] For an approach to stress learning that involves constraints that are not pre-specified, see Hayes and Wilson (2008).

light syllables and *L1* representing a syllable with primary stress.

(1) /L L L/ → [L **L1** L]

This datum illustrates the ambiguity present when learning stress patterns like these in a foot-based theory—this word could contain a left-aligned foot with iambic stress like [(L L1) L] or a right-aligned foot with trochaic stress like [L (L1 L)]. Each of the 124 languages consists of 62 mappings like this. The 62 input strings are all possible combinations of *L*s and *H*s for strings of 2 to 5 syllables in length, plus strings of 6 and 7 *L*s. For each input string, the candidate set of output strings consists of all possible parsings of the syllables into unary and binary feet, including ones where syllables are left unparsed. There is a minimum of one foot for each word. One of the stresses is designated as primary, and the candidate set has all possible primary stress placements.

Each output string has a corresponding vector of constraint violations, for the 12 constraints shown in (2). Each of the 124 languages in the test set can be generated by some OT ranking of these constraints. That is, some ranking can make a parsed structure optimal that is consistent with the stress pattern in the output of the learning datum, for all 62 target mappings.

(2) Constraints from the TS Data Set (constraint definitions from Jarosz, 2013)

a. *FtBin*: Each foot must be either bimoraic or disyllabic.
b. *Parse*: Each syllable must be footed.
c. *Iambic*: The final syllable of a foot must be the head.
d. *FootNon-fin*: A head syllable must not be final in its foot.
e. *Non-fin*: The final syllable of a word must not be footed.
f. *WSP*: Each heavy syllable must be stressed.
g. *WordFoot-R*: Align right edge of the word with a foot.
h. *WordFoot-L*: Align left edge of the word with a foot.
i. *Main-R*: Align head foot with right edge of the word.
j. *Main-L*: Align head foot with left edge of the word.
k. *AllFeet-R*: Align each foot with right edge of the word.
l. *AllFeet-L*: Align each foot with left edge of the word.

TS proposed that a learner uses its current grammar to parse a form and then updates its constraint rankings according to that parse. Most subsequent work (with the exception of Jarosz, 2015) has been based on this general premise (Jarosz, 2013; Boersma and Pater, 2016).

TS found that when they ran their model 10 times on each of the 124 languages in the data set, it achieved perfect accuracy on a language 60.48% of the time. Boersma and Pater (2016) found that when they used a similar parsing strategy, but with numerically weighted constraints instead of ranked ones, and with a stochastic component in the parsing process, languages were learned fully correctly 88.63% of the time. Jarosz (2013) pushed performance on this data set even further, showing that by revising the parsing strategy, success could be achieved 94.19% of the time over 10 runs of the 124 TS languages.

The state-of-the-art on the TS data for constraint-based models (95.73%) was achieved by Jarosz (2015) whose model used a pair-wise ranking grammar with a learning algorithm inspired by expectation maximization (Dempster et al., 1977). This allowed the model to avoid the problem of parsing altogether, since it was able to sample the mappings that various constraint rankings create over the course of acquisition to see which were most likely to improve its performance.

3 Our Model

While various neural network architectures have been used in phonology, such as feedforward networks (e.g., Gupta and Touretzky, 1994; Moreton, 2012), simple recurrent networks (e.g., Hare, 1990), and convolutional neural networks (e.g., Beguš, 2020), here we focus on the sequence-to-sequence architecture (Seq2Seq Sutskever et al., 2014).

This architecture was originally constructed for machine translation, but is convenient for modeling phonological mappings since it can straightforwardly map between strings of differing lengths, needed for dealing with processes like epenthesis and deletion. This is accomplished by processing the input and output strings with separate recurrent neural networks. The input is fed into the first network (called the *encoder*), which has no output layer. The recurrent connections of the encoder then pass information about the input to the second network (called the *decoder*), which has no input, but *does* have an output layer.

A number of studies have shown that when applied to phonological patterns, Seq2Seq networks display similar learning biases to humans (Prickett, 2019, 2021) and also generalize in a human-like way on phonological and morphological tasks (Kirov and Cotterell, 2018; Prickett et al., 2018); see Corkery et al. (2019) for some caveats.

In this paper, we test the Seq2Seq architecture with both GRU (Cho et al., 2014) and LSTM (Bengio et al., 1994) layers. While both were created to help recurrent networks learn longer dependencies (specifically by addressing the problem of *vanishing gradients*; Bengio et al., 1994), past work has found that some differences exist in the biases each kind of layer has. For example, GRU layers have been shown to be biased against learning counting-based patterns that LSTMs easily acquire (Weiss et al., 2018).

In all of the simulations presented here, the network had 2 layers each in its encoder and decoder, with 20 units in each layer, and hyperbolic tangent activation functions throughout. The learning algorithm Adam (Kingma and Ba, 2015), with a batch size of 1, was used to minimize the mean squared error between the model's output and the correct output throughout learning. We leave performing a proper grid search to determine how well our results generalize to other hyperparameter settings to future work.

4 Methods and results[2]

4.1 Original Tesar and Smolensky (2000) Languages

We first tested our model to see how well it could learn the 124 original languages in the TS data set. In each input string, a timestep for the model represented a single syllable, with a [syllable weight] feature distinguishing between light ($= -1$) and heavy ($= 1$) syllables. In the output, timesteps again represented individual syllables, with the features [stress] and [primary] used to distinguish between syllables with primary stress (values of 1 and 1, respectively), secondary stress (values of 1 and -1, respectively), and no stress (values of -1 and -1, respectively).

We ran the model with a learning rate of .0005 for 500 epochs once on each of the 124 languages in the TS set. We tried versions of the Seq2Seq architecture with both GRU and LSTM layers in them and found that both layer types achieved perfect accuracy[3] in 98.39% (122/124) of the languages. This represents the highest rate of success for any model on this test set. However, it's unclear whether the model was actually encoding generalizable information, or just memorizing the 62 mappings present in each language, which would be a fairly trivial task.

Previous research has used the constraints in (2), which are all defined to hold for any string of a particular phonological type. They provide no way of encoding a situation in which two strings of a given type behave differently, as occurs in many real languages (in "exceptions", or more generally in lexically conditioned patterns). The acquired constraint-based grammars are therefore guaranteed to generalize, though at the cost of not being able to capture lexically conditioned patterns. In what follows, we test whether our learner does learn generalizable representations of the data by including multiple tokens of each type of input string. We then turn to the question of whether it can learn lexically conditioned stress patterns, including restrictions on the distribution of lexical stress.

4.2 Generalization from Tesar and Smolensky (2000) Languages

To test whether the Seq2Seq network was learning generalizable patterns or just memorizing the mappings in each language, we introduced an extra set of "lexical" features to the inputs of the TS data set. These features were implemented as a random, base-2 label for each of the tokens in training, representing the different tokens of each mapping that one would expect in an actual language. For example, the mapping from (1) would have multiple copies in training, each of which had a unique, non-zero label in their input, as illustrated in (3). These are meant to represent multiple words in a language with three light syllables and penultimate stress (like English *banana* and *cabana*).

[2] For the software used in the simulations presented here, see https://github.com/blprickett/Neural-Network-Stress.

[3] Since the network's output takes the form of a vector of real-numbered feature values, each mapping was considered correct if every feature in every timestep of its output had the correct sign (positive or negative), given that mapping's input.

Table 1: Percent of languages with perfect accuracy in training and testing.

Tokens per Type	Training	Testing
3	86.29	44.35
6	98.39	90.32

(3) Examples of Multiple Tokens of the Same Mapping Type

a. /L L L/$_{0101}$ → [L **L1** L]
b. /L L L/$_{1111}$ → [L **L1** L]
c. /L L L/$_{0001}$ → [L **L1** L]

Crucially, each token that belonged to the same mapping type had the same output in all of these simulations (that is, there was no lexical conditioning in any of the stress patterns). We created two sets of data using this system: one that had 3 tokens for each of the 62 mapping types in each of the TS languages and one that had 6 tokens for each of the types.

We ran the GRU version of the Seq2Seq model on these two data sets with a learning rate of .0005 for 200 epochs. At the end of training, we tested the model on 62 novel pieces of data, each of which represented one of the mapping types from training, but with only 0's for the lexical label features. If the network learned a generalizable pattern from training, it should correctly map all 62 of the novel testing items. The results for training and testing on both data sets are shown in Table 1.

These results suggest that, with enough tokens per type, the Seq2Seq network *does* generalize correctly from almost all (112 out of 124) of the languages in the TS data. Additionally, the accuracy on both training and testing increased with the number of tokens per type, suggesting that a number of tokens higher than 6 might allow the model to do even better on both. Natural languages of course tend to have more than six words with a given type of stress pattern, at least for shorter words.

4.3 Languages with Lexically Conditioned Stress

The final test of our model did not directly use any of the languages from the TS data set. Instead, we used the 62 input syllable strings from the TS languages and created output stressings for them using 6 novel patterns. These patterns involved *stress windows* (Kager, 2012), meaning they allowed stress to appear on any of a set of contiguous syllables at the word edge in the output, with the syllable that's stressed in a specific word being lexically specified.

Our patterns involved two basic types of window: right aligned and left aligned. Each pattern had windows of size 2, meaning the right aligned languages always had stress on their ultimate or penultimate syllables and the left aligned languages always had stress on the first or second syllables. The other feature that varied across languages was how likely stress was to occur on each of the two syllables in a window. We created three conditions for this variable: languages in which the first syllable of a window was stressed 25% of the time and second was stressed 75%, languages in which both syllables in the window were equally likely to be stressed, and languages in which the first syllable of a window was stressed in 75% of words and the second was stressed in 25% of them.

These two variables created 6 total languages to test the model on. In every language, there were 4 tokens for each mapping type, with the proportion of first syllable/second syllable stress in types' windows being the same as the language itself. This is illustrated in (4) for the language with left-aligned windows and stress on the first syllable of the window in 25% of words.

(4) Examples of Stress Window Data

a. /L L L L/$_{0101}$ → [**L1** L L L]
b. /L L L L/$_{1111}$ → [L **L1** L L]
c. /L L L L/$_{0001}$ → [L **L1** L L]
d. /L L L L/$_{1001}$ → [L **L1** L L]

We trained the model ten times on each of these 6 languages, with a learning rate of .005, until the model reached perfect accuracy on the training data. The LSTM model was able to reach this criterion for all 6 languages, while the GRU was unable to reach it for any of them in a reasonable number of epochs (we tried a variety of values for this, running the GRU model for up to 10,000 epochs with no success). At the end of training, we tested the model on novel data that had values of zero for all of the lexical label features to see how it generalized these lexically specified patterns. Table 2 shows the results on testing data for the LSTM model (no GRU results are shown since that model never succeeded in training).

With the exception of the language with left aligned windows and stress on the second syllable 25% of the time, the model seems to generalize to novel data in a way that reflects the statistics of

Table 2: Proportion of words in each language in which the window's second syllable is stressed.

Edge of the Word	Prob. in Training	Model's Results on Testing (SD)
Left	.75	0.874 (0.15)
Right	.75	0.749 (0.22)
Left	.5	0.706 (0.23)
Right	.5	0.554 (0.27)
Left	.25	0.123 (0.19)
Right	.25	0.332 (0.26)

the language it was trained on (with perhaps a bias toward stressing the second syllable more often). These results suggest that the LSTM model not only successfully learns these patterns that involve lexically conditioned stress but also can keep track of general statistical trends in the language, as has been experimentally documented for humans (see, e.g., Ernestus and Baayen, 2003).

5 Discussion

5.1 Comparison with earlier research

Our results on the TS data set with a Seq2Seq model are comparable to the best achieved with constraint based models. It is difficult to compare directly, since the 98.39% accuracy achieved in the first set of simulations, as well as on the training data in the 6 lexical item condition, could be attributable to the model simply representing each of the individual mappings, rather than learning generalizable representations. Nonetheless, the fact that it generated the correct stress pattern for 90.32% of the unseen tokens when there were 6 tokens of each type in the training data shows that it is capable of learning these patterns in a generalizable way with a high degree of accuracy.

None of the prior research on the TS data set provided a means for representing lexical idiosyncrasy, cases where two words of the same syllable shape have different stress patterns. There is a body of prior work on constraint-based approaches to lexical idiosyncratic phonology, however. Tesar (2006) presents an approach to learning exceptions in terms of contrastive specification of underlying features, Pater et al. (2012) propose an alternative that uses constraints on Underlying Representations within a MaxEnt learning framework, Moore-Cantwell and Pater (2016) explore the use of lexically specific constraints in MaxEnt, Hughto et al. (2019) study similar lexically scaled constraints, and Nazarov (2018) presents another approach to learning lexically specific constraints. All of this work has been done on very small systems, and it is not immediately clear how well the proposals will scale up to cases with even the number of constraints in the TS test set, let alone constraint sets that are large enough to deal with the complexity of less idealized individual languages, and of a fuller typology.

5.2 Future Work

A number of avenues exist for future work. The results presented in §4.2 made the TS data set more realistic by introducing multiple lexical tokens for each type of mapping. Making the TS data even more realistic is one potential future direction, for example, by representing inputs and outputs as strings of phonemes rather than just strings of light and heavy syllables.

Another question to investigate is how well this model and previous computational models of stress deal with other patterns involving exceptionality. The stress window languages introduced in 4.3 are a step in this direction, but more complex patterns of lexically conditioned stress could be explored. The constraint-based models previously tested on the TS data set had no way to represent lexical information, so equipping these simpler models with a way to handle such patterns (with, e.g., *lexically indexed constraints*; Pater, 2009) could also be fruitful.

A limitation of the TS data set is that it is based on a factorial typology of constraints rather than a real-world typology of stress-based patterns. Future work should sort through these artificially constructed languages to see which of them have real-world counterparts and which are unattested. At that point, looking closer at the learning difficulty across languages might help to explain why some are absent from the typology. Gupta and Touretzky (1994) provide an analysis of their learning results with a neural model that gives an example of how this research could proceed.

Finally, computational phonology often involves comparing predictions made by models to human behavior in artificial language learning studies (e.g., Wilson, 2006). Such studies involving stress patterns *do* exist (e.g., Carpenter, 2016), and future work should compare the aquisition and generalization observed in them to that of computational models of stress learning.

5.3 Conclusions

In this paper, we presented results showing that a Seq2Seq neural network can successfully learn a variety of stress patterns. Using the Tesar and Smolensky (2000) data set (a commonly cited benchmark for models of stress), we were able to show that the network outperformed past models when tested on how many of the languages in the data set it could acquire perfectly.

We then created an extension of this data set that included multiple tokens of each relevant mapping type in the 124 languages, and differentiated these tokens using lexically specific labels for each word. When the model was given data that included six tokens for each mapping type from the original data set, its performance on novel test items was comparable to past, state-of-the-art approaches.

Finally, we showed that the LSTM-based model could successfully learn lexically-conditioned patterns involving stress windows (Kager, 2012), something that past constraint-based models of hidden structure do not have the expressive power to do.

Taken together, these results show that (i) prespecified constraints are not necessary for a model to succesfully learn and generalize stress-based patterns and (ii) while the neural network we used had the ability to simply memorize the mappings we were training it on, it instead learned a general pattern for most languages that could be applied to novel forms.

Acknowledgements

We would like to thank the UMass Sound Workshop, as well as the audiences of the 2019 Manchester Phonology Meeting and the 2021 Annual Meeting on Phonology for helpful discussion of topics related to this paper. This research was supported by the National Science Foundation grant BCS 1650957.

References

Gašper Beguš. 2020. Modeling unsupervised phonetic and phonological learning in generative adversarial phonology. *Proceedings of the Society for Computation in Linguistics*, 3(1):138–148.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Paul Boersma and Joe Pater. 2016. Convergence properties of a gradual learner in Harmonic Grammar. In John J. McCarthy and Joe Pater, editors, *Harmonic Grammar and Harmonic Serialism*, pages 389–434. Equinox Publishing, Bristol, Connecticut.

Angela C Carpenter. 2016. The role of a domain-specific language mechanism in learning natural and unnatural stress. *Open Linguistics*, 2(1).

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? Encoder-decoder neural networks as cognitive models of English past tense inflection. *arXiv:1906.01280 [cs]*. ArXiv: 1906.01280.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

B. Elan Dresher and Jonathan D. Kaye. 1990. A computational learning model for metrical phonology. *Cognition*, 34(2):137–195.

Mirjam Ernestus and R Harald Baayen. 2003. Predicting the unpredictable: Interpreting neutralized segments in dutch. *Language*, pages 5–38.

Prahlad Gupta and David S. Touretzky. 1994. Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive science*, 18(1):1–50.

Mary Hare. 1990. The role of trigger-target similarity in the vowel harmony process. In *Annual Meeting of the Berkeley Linguistics Society*, volume 16, pages 140–152.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Coral Hughto, Andrew Lamont, Brandon Prickett, and Gaja Jarosz. 2019. Learning Exceptionality and Variation with Lexically Scaled MaxEnt. Publisher: University of Massachusetts Amherst.

Gaja Jarosz. 2013. Learning with hidden structure in optimality theory and harmonic grammar: Beyond robust interpretive parsing. *Phonology*, 30(1):27–71.

Gaja Jarosz. 2015. Expectation driven learning of phonology. *Ms., University of Massachusetts Amherst*.

René Kager. 2012. Stress in windows: Language typology and factorial typology. *Lingua*, 122(13):1454–1493.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, Conference Track Proceedings*.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Claire Moore-Cantwell and Joe Pater. 2016. Gradient Exceptionality in Maximum Entropy Grammar with Lexically Specific Constraints. *Catalan Journal of Linguistics*, 15:53.

Elliott Moreton. 2012. Inter-and intra-dimensional dependencies in implicit phonotactic learning. *Journal of Memory and Language*, 67(1):165–183.

Aleksei Nazarov. 2018. Learning within- and between-word variation in probabilistic OT grammars. *Proceedings of the Annual Meetings on Phonology*, 5.

Joe Pater. 2009. Morpheme-specific phonology: Constraint indexation and inconsistency resolution. In Steve Parker, editor, *Phonological Argumentation: Essays on Evidence and Motivation*. Equinox, London.

Joe Pater, Robert Staubs, Karen Jesney, and Brian Smith. 2012. Learning probabilities over underlying representations. In *Proceedings of the Twelfth Meeting of the ACL-SIGMORPHON: Computational Research in Phonetics, Phonology, and Morphology*, pages 62–71.

Brandon Prickett. 2019. Learning biases in opaque interactions. *Phonology*, 36(4):627–653.

Brandon Prickett. 2021. Modelling a subregular bias in phonological learning with recurrent neural networks. *Journal of Language Modelling*, 9(1).

Brandon Prickett, Aaron Traylor, and Joe Pater. 2018. Seq2seq Models with Dropout can Learn Generalizable Reduplication. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 93–100.

Alan Prince and Paul Smolensky. 2004. *Optimality Theory: Constraint interaction in generative grammar*. Blackwell.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Bruce Tesar. 2006. Faithful Contrastive Features in Learning. *Cognitive Science*, 30(5):863–903.

Bruce Tesar and Paul Smolensky. 2000. *Learnability in optimality theory*. Mit Press.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision rnns for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745.

Colin Wilson. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive science*, 30(5):945–982.

Linguistic Complexity and Planning Effects on Word Duration in Hindi Read Aloud Speech

Sidharth Ranjan
IIT Delhi
sidharth.ranjan03@gmail.com

Rajakrishnan Rajkumar
IISER Bhopal
rajak@iiserb.ac.in

Sumeet Agarwal
IIT Delhi
sumeet@iitd.ac.in

Abstract

Our study investigates the impact of linguistic complexity and planning on word durations in Hindi read aloud speech. Reading aloud involves both comprehension and production processes, and we use measures defined by two influential theories of sentence comprehension, *Surprisal Theory* and *Dependency Locality Theory*, to model the time taken to enunciate individual words. We model planning processes using an information-theoretic measure we call FORWARD SURPRISAL, inspired by surprisal theory which has been prominent in recent psycholinguistic work. Forward surprisal aims to capture articulatory planning when readers incorporate parafoveal viewing during reading aloud. Using a Linear Mixed Model containing memory and surprisal costs as predictors of word duration in read aloud speech (parts-of-speech and speakers being intercept terms), we investigate the following hypotheses: 1. High values of linguistic complexity measures (lexical+PCFG surprisal and DLT memory costs) lead to high word durations. 2. High values of forward lexical surprisal tend to induce high word durations. 3. High-frequency words are read aloud faster than low-frequency words. We validate the above hypotheses using data from the TDIL corpus of read aloud speech. Further, using a Generalized Linear Model to predict content and function word labels we show that lexical surprisal measures do not help distinguish between these 2 classes. Thus reading aloud might not involve distinct access strategies for content and function words, unlike spontaneous speech.

1 Introduction

Prior work on language production (Ganushchak and Chen, 2016; Navarrete et al., 2016) presents a long-standing debate on the cognitive processes involved in *spontaneous speech* and *reading aloud*. Although both the modalities deal with language production, their unifying accounts have been underexplored in the literature (Sulpizio and Kinoshita, 2016). Spontaneous speech involves the packaging of non-linear conceptual information into linear (sequential) ordering of words in a sentence. In this process, speakers optimize for words, syntactic alternations, and memory load (Slevc, 2011). On the contrary, the cognitive mechanism in reading aloud involves a two-step process, namely *word recognition* and *articulation*. Therefore, various representational levels of words, such as orthographic, phonological, phonemic, and visual information interact with on another to generate the pronunciation of a word.

Motivated by a long of line of previous work in both traditions, our current study investigates the relationship of word duration with linguistic complexity and planning effects in Hindi read aloud speech. To this end, we quantified linguistic complexity using contextual predictability measures defined by *Surprisal Theory* (Hale, 2001; Levy, 2008) and memory costs stipulated by Dependency Locality Theory (DLT, Gibson, 2000). Although surprisal and DLT measures were originally proposed for language comprehension, recent work points towards their efficacy in modelling language production. Mathematically, surprisal is the same as information density. Jaeger (2010) showed that the realization of the optional *that*-complementizer in English spontaneous speech is influenced by uniform information density considerations. Moreover, predictable words tend to be spoken fast (Bell et al., 2003) with reduced emphasis on fine-grained acoustic details (Pluymaekers et al., 2005). In order to investigate planning effects, we used the model-

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 119-132.
Held on-line February 7-9, 2022

ing framework proposed by Bell et al. (2009) for spontaneous speech and adapted their following bigram probability measure to capture *production planning* when reading aloud. We investigated 3 hypotheses using Linear Mixed Models (LMMs, Pinheiro and Bates, 2000) containing all the above measures and low-level predictors generally used in previous work (word frequency and length) to predict word durations (parts-of-speech and speakers being intercept terms). Our hypotheses and their motivation are provided below :

1. *High values of linguistic complexity measures (lexical+PCFG surprisal and DLT integration+storage costs) lead to high word durations*: Researchers have shown that such complexity measures account for production difficulties as well, such as disfluencies (Scontras et al., 2014; Dammalapati et al., 2021) and word duration (Demberg et al., 2012) in spontaneous speech.

2. *High values of forward lexical surprisal tend to induce high word durations*: We deployed a measure named *forward surprisal*, inspired from *Surprisal Theory*) and originally proposed by Ranjan et al. (2020). Cognitively, this measure (negative log probability of a word given upcoming words) models parafoveal preview in the reading part of reading aloud, and thus such look-ahead helps in articulatory planning during subsequent production processes.

3. *High-frequency words are read aloud faster than low-frequency words*: The *Dual Route Cascaded* model (Coltheart et al., 2001, DRC) of word recognition and reading aloud predicted and demonstrated this for isolated single words by means of lexical decision and reading aloud tasks.

All the above hypotheses were validated in our experiments conducted on the publicly available TDIL corpus of read-aloud Hindi speech. Forward surprisal is a significant positive predictor of word durations even in the presence of other factors, pointing towards planning effects in reading aloud. High values of trigram lexical surprisal and PCFG syntactic surprisal along with DLT storage costs induced high word durations. For English spontaneous speech, Bell et al. (2009) revealed asymmetric behavior of lexical predictability measures on function vs. content word duration. They attributed this finding to differences in how content and function words are accessed in the mind (*i.e.*., lexical access during spontaneous speech) apart from their properties pertaining to grammatical function. For *reading aloud* Hindi speech data, we found that lexical predictability of both content and function words have identical effects in predicting reading aloud times. An increase in both backward and forward surprisal measures of lexical surprisal led to identical effects on word durations (*i.e.*, increased durations) of both content and function words in read aloud speech. Going beyond Bell et al. (2009), for the separate task of predicting *content* and *function* class labels for each word using a Generalized Linear Model, we showed that trigram lexical surprisal measures are not significant predictors of word class. In contrast, PCFG surprisal induced a significant boost in prediction accuracy for this task. Thus we found differential effects of lexical and surprisal measures in reading aloud.

Our main contribution is that we extend the prior work motivating our hypotheses (as cited above) by validating them in the presence of a comprehensive host of factors in a language other than English. To the best of our knowledge, this is the first work that explores reading aloud production times in Hindi. Both Ranjan et al. (2020) and Demberg et al. (2012) did not incorporate DLT-based predictors, while the former work did not include syntactic surprisal in their regression models. Scontras et al. (2014) did not factor in surprisal-based factors in their spontaneous production experiments on relative clauses. Finally, the DRC model motivating the third hypothesis above deals with the recognition and production of isolated words. In this work, we extend its prediction to entire sentences. Based on the identical effects of both forward and backward lexical surprisal measures, we offer preliminary evidence that lexical access of items to the extent of the full semantic representation of a word may not be necessary during reading aloud processes. This finding is compatible with the DRC model assumption of word processing via the non-semantic lexical route.

The paper is structured as follows. Section 2 provides background on theories and models pertaining to this work. Section 3 presents the details about the dataset and methods used in this work. Section 4 illustrates our main experiments and their results. Section 5 summarizes our main findings and discusses their implications along with pointers to future work.

2 Background

The following subsections provide essential background on the Hindi language and its orthography, the Dual Route Cascaded (DRC) model, Dependency Locality Theory, and Surprisal Theory.

2.1 Hindi Language and Script

Hindi is a head-final language with relatively free word order (with Subject-Object-Verb being the canonical order) compared to English, and has a rich case-marking system realized as postpositions (Agnihotri, 2007). Hindi adopts the Devanagari alphasyllabary-based writing system. The Devanagari script is composed of 47 characters containing 33 consonants (क, ख, ग, etc.) and 14 vowels (अ, आ, इ, etc.). In terms of letter-sound correspondence, the orthography of the script mostly corresponds with grapheme pronunciation except for cases when vowel diacritics, conjunct consonants or ligatures are present (Vaid and Gupta, 2002). Further details of the script are provided in Appendix C.

2.2 Dual Route Cascaded (DRC) Model

The DRC model is a computational model of the *visual word recognition* and *reading aloud*. The model posits two separate cognitive routes i.e., *lexical* and *sub-lexical* that are involved in reading aloud, and within each route, the information processing occurs in a cascaded fashion (Coltheart et al., 2001). It is a computational implementation of the dual-route theory of reading and further stipulates three routes for word processing, *viz.* Grapheme-Phoneme Correspondence (GPC) route, Lexical Semantic route and Lexical Nonsemantic route. Figure 5 in Appendix B provides a visual illustration of the DRC model. Empirical evidence for the efficacy of the DRC model emerges from its ability to simulate human latencies in the tasks of reading aloud and lexical decision tasks. DRC adapts the rationale for *frequency effects* from earlier work on word processing. Morton (1969) demonstrated that high frequency words required lower evidence from visual input (*i.e.*, letters in reading) on account of their lower activation. Subsequently, word naming occurs on account of a lexical search procedure (Forster and Chambers, 1973) where activation levels affect search latencies.

2.3 Dependency Locality Theory

Dependency Locality Theory is a theory of sentence comprehension proposed by Gibson (2000) which posits two processing costs at each word, *viz*, INTEGRATION and STORAGE COSTS (defined and exemplified in Section 3). DLT predictions about the increased comprehension difficulty of object relative clauses over subject relative clauses have been validated using per-word reading time data in a variety of languages. Scontras et al. (2014) showed that object relative clauses are harder to produce than subject relative clauses and relative clause production times are connected to DLT-based memory costs. For Hindi, the eye-tracking based reading times in comprehension have been known to be influenced by DLT-inspired costs (Husain et al., 2015; Agrawal et al., 2017).

2.4 Surprisal Theory

Surprisal Theory (Hale, 2001; Levy, 2008) posits that comprehenders construct probabilistic knowledge based on previously encountered structures. Mathematically, *surprisal* of the $(k+1)^{th}$ word, w_{k+1}, is defined as negative logarithm of conditional probability of word, w_{k+1} given the preceding context which can be either sequence of words or a syntactic tree:

$$S_{k+1} = -\log P(w_{k+1}|w_{1...k}) = \log \frac{P(w_1...w_k)}{P(w_1...w_{k+1})} \quad (1)$$

Both the versions of surprisal i.e., *lexical and syntactic* configurations have been shown to account for eye-movements reading (Demberg and Keller, 2008; Agrawal et al., 2017; Staub, 2015) as well as self-paced reading time data (Smith and Levy, 2013). Pioneering work by Demberg et al. (2012) showed that both n-gram and PCFG-based syntactic surprisal measures were significant positive predictors of word duration in spontaneous speech. More recently, Dammalapati et al. (2021) demonstrated that surprisal and DLT-based metrics predict speech disfluency using English spontaneous speech corpus.

3 Data and Methods

Our dataset consists of 1531 sentences (from scientific and technical genre) from the TDIL corpus of Hindi read aloud speech[1]. One male and one female speaker were asked to record their speech by reading aloud 341 sentences (4,444 words) and 1,190 sentences (11,163 words), respectively. Table 4 in Appendix C illustrates pertinent word-level properties (overall and grammatical category-wise). Word durations were extracted from the recorded speech using the PRAAT software package. We estimated various word-level cognitive measures as described below:

1. **Word length:** Total number of consonants and vowels present in the word (इसलिए – *isliye; therefore* has word length of 4; 2 consonants (स, ल) and 2 vowels (इ, ए).

2. **Word frequency**: Count of each target word as obtained from the EMILLE Hindi corpus (Baker et al., 2002).

3. **Unigram surprisal:** Negative log probability of individual target word.

4. **Backward surprisal:** Negative log of probability of target word given two preceding words in the context (Equation 1).

5. **Forward surprisal:** Negative log of probability of target word given two following words in the context. So the surprisal of the k^{th} word is estimated as: $S_k = -\log P(w_k \mid w_{k+1}, w_{k+2})$

6. **PCFG surprisal:** Negative log probability of target word given contextual syntactic tree (Equation 1).

7. **Integration cost (IC)**: Backward looking cost denoting the sum of distances between the word to be integrated into the structure processed so far and its previous heads/dependents. Distance is the number of intervening words between each head and dependent.

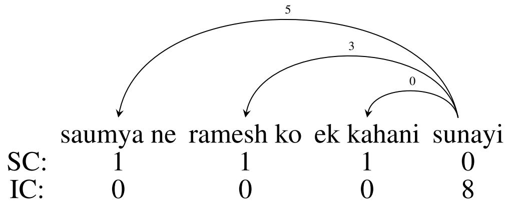


Figure 1: Integration and storage cost calculations for the sentence '*Saumya narrated a story to Ramesh*', with head-dependent distance indicated above each dependency link; example sentence adapted from Husain et al. (2015)

8. **Storage cost (SC)**: Forward-looking cost corresponding to the number of incomplete dependencies in the upcoming structure.

Unigram and Trigram Surprisal measures for each word in a sentence was computed using unigram and trigram language models respectively trained on the EMILLE corpus of written text with mixed genre (Baker et al., 2002) using the SRILM toolkit (Stolcke, 2002) with Good-Turing discounting smoothing algorithm. **PCFG surprisal** for each word was estimated by training an incremental probabilistic left-corner parser (van Schijndel et al., 2013) on 13,000 phrase structure trees (converted from HUTB dependency trees) using ModelBlocks toolkit[2] (Refer Appendix D for more details on training data and settings). We calculated DLT IC and SC costs automatically following the definitions adopted by Husain et al. (2015). See Figure 1 for an illustration. They computed DLT costs by hand for a small corpus, while our DLT SC and IC costs were computed from dependency trees obtained by parsing TDIL sentences using the ISC dependency parser[3] (Bhat, 2017) trained on HUTB gold standard dependency trees (parser performance documented by Bhat: UAS of 93.52% and a LAS of 87.77%).

4 Experiments and Results

In the following subsections we describe the specific experiments and results of this study.

4.1 Correlation Results

Prior to performing the regression experiments described in the next few subsections, we computed

[1] https://tdil-dc.in

[2] https://github.com/modelblocks

[3] https://bitbucket.org/account/user/iscnlp/projects/ISCNLP

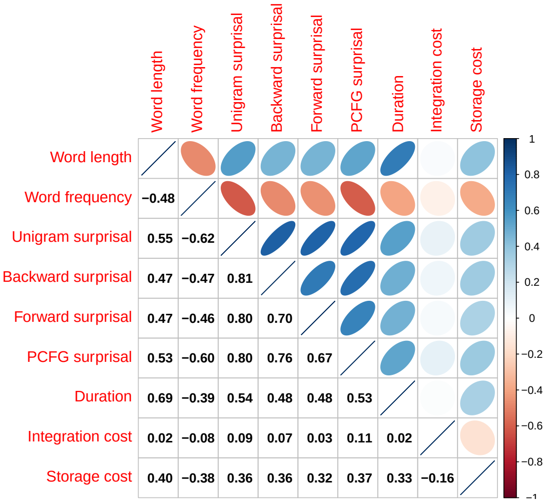


Figure 2: Pearson's correlation coefficients amongst the different predictors and word duration

Predictors	**Estimate**	**Std. Error**	**t-value**
Intercept	5.525	0.098	56.364
Word length	0.217	0.003	62.430
Unigram surprisal	0.027	0.006	4.284
Word frequency	-0.034	0.004	-7.643
SC	0.010	0.004	2.309
IC	-0.016	0.003	-5.830
Backward 3g-surprisal	0.015	0.005	3.128
Forward 3g-surprisal	0.032	0.004	7.181
PCFG surprisal	0.051	0.005	10.412

Table 1: Fixed effects of an LMM predicting reading aloud time (15607 data points; all predictors are significant for the |t|=2 threshold)

the Pearson's coefficient of correlation between the different predictors. We also computed the correlation between each predictor and the dependent variable, word duration. Figure 2 displays the correlation results. The high positive correlation between word duration and all surprisal scores suggests that the words which are easy to produce by virtue of high predictability in context tend to have lower reading time and vice versa. DLT-storage costs display low correlation with other predictors, while integration cost shows negligible correlation with any other predictor, indicating their independent impact. SC and IC costs show low negative correlation with one another as they are forward and backward-looking costs respectively and thus might work differently. We also observe that word length is highly correlated with word duration as is observed in previous production (Bell et al., 2009) and comprehension studies (Husain et al., 2015; Agrawal et al., 2017).

4.2 Regression Experiments

We trained Linear Mixed Models (LMMs) to predict per-word duration (transformed to a logarithmic scale following previous work). The logarithmic scaling of the independent variables, *viz.* surprisal measures, took care of highly varied frequencies during model training. All the independent variables were normalised to z-scores, *i.e.*, the predictor's value (centered around its mean) was divided by its standard deviation. We have used the Glm package in R to perform our regression experiments using a very basic model, given below in R GLM format (independent variable $\sim$ dependent variables + 1| random intercept terms):

$$Duration \sim word\ length + word\ frequency + unigram\ surprisal + backward\ surprisal + forward\ surprisal + PCFG\ surprisal + IC + SC + 1|Speaker + 1|POS$$

The POS intercepts were based on tags obtained by converting HUTB POS tags to 11 universal POS tags corresponding to content words (verb, noun, adjective, and adverb) as well function words (postposition, pronoun, determiner, particle, conjunction, question, and quantifier).

Our regression results documented in Table 1 reveal that all the measures are significant in predicting the read-aloud word duration and their regression coefficients are in the expected direction, thus validating our original hypotheses stated in Section 1. Frequency and unigram surprisal capture the frequency and predictability effects of individual words, *i.e.*, frequent words require less time and effort to activate phonemes for articulation (as predicted by the DRC model). The positive coefficients of all surprisal and DLT SC measures show that with an increase in each predictor's value, the word duration in read-aloud speech increases. However, DLT IC has an unexpected negative coefficient, an anomaly which has been also reported in the comprehension literature (Demberg and Keller, 2008; Husain et al., 2015). Demberg and Keller (2008) analyzed this anomaly rigorously and showed that in the presence of other predictors, integration cost works in the expected direction (*i.e.*, high integration costs induce high reading times) only for higher range IC values. Future inquiries need to examine whether this result carries over to the production setting and the implications of such a finding for integrated models of both processes (a theme we take up at the end of Section 5). In the following subsections, we now discuss the impact of selected measures on reading aloud word duration.

4.2.1 Forward Surprisal

The positive regression coefficient of forward surprisal (Table 1) suggests that the difficulty associated with the upcoming words has a role in determining the reading time of the current word. The effect of forward surprisal on duration is illustrated using the following examples (region of interest: *vidyalaye*; *school*):

1. pahle pitaji bacchon=ko **vidyalaye=se** lene jaate the
   before father child=ACC school=ABL take go be-PST.3SG
   Earlier father used to take children from school

2. bacche **vidyalaye=se** aate hi khelne chale gaye
   children school=ABL come EMPH play go-PST.PL
   The children went to play as soon as they came from school

In the first example above, the word *vidyalaye* (550ms duration; 4.55 forward surprisal) has a higher surprisal and longer duration compared to the same word in the second sentence (510ms; 3.90 bits). This is because *vidyalaye se aate* is a much more frequent sequence than *vidyalaye se lene* in the trigram training corpus. Thus planning effects are modelled by this measure, a theme we explore in the next subsection.

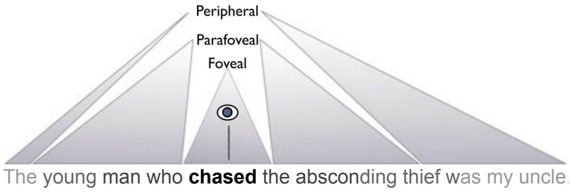

Peripheral
Parafoveal
Foveal
The young man who **chased** the absconding thief was my uncle.

Figure 3: Parafoveal preview in reading; adapted from Schotter et al. (2012)

Interactions	Estimate	Std. Error	t-value
MODEL 1			
Word length x Backward 3g-surp	**-0.024**	0.004	-5.491
Word length x Forward 3g-surp	**-0.031**	0.004	-8.061
Word length x PCFG surprisal	0.001	0.005	0.314
MODEL 2			
Function word x Backward 3g-surp	**0.028**	0.009	2.936
Function word x Forward 3g-surp	**0.041**	0.008	4.855
Function word x PCFG surprisal	**-0.039**	0.009	-3.953

Table 2: Two different LMMs displaying only the interaction terms of surprisal with word length (top) and function word (bottom) respectively predicting reading aloud time; see full model results in Appendix E (15607 data points; all significant predictors denoted by |t|>2)

4.2.2 Parafoveal Preview and Word Length Effects

It is well understood that the length of words influences the reader's eye movements as long words induce more fixations of greater duration than short words (Just and Carpenter, 1980; Rayner et al., 1996). In this context, Bicknell and Levy (2012) argue that uncertainty about the length of words affects the word reading duration. They posit that the uncertainty increases proportionally with an increase in word length, leading to more fixation and longer word duration. We hypothesize that if the forward surprisal effect is driven by parafoveal previewing (as illustrated in Figure 3), there should be smaller predictability effects with longer target words. This is because longer target words will lead to less linguistic material visible in the parafoveal region, thus not allowing for informative computation of the target word's forward surprisal. We investigated the effect of word length on word duration using another linear mixed model containing word length and surprisal interaction terms. Table 2 (top block) documents the interaction results, which show that the effect of forward trigram surprisal on reading-aloud times decreases by 0.02 with every unit increase in the word length, thus confirming our hypothesis. A similar result is obtained in case of backward trigram surprisal as well. See Table 5 in Appendix E for full regression model results. The relative strengths of forward and backward surprisal measures in both production and comprehension needs to be systematically investigated in future inquiries.

4.2.3 Word Class and Duration

This section and the next one are motivated by the findings of Bell et al. (2009). For spontaneous speech, they showed that both function and content word duration were significantly predicted by the following word (*forward probability*). However, unlike content words, function word duration was determined significantly by the previous word only (*backward probability*). Content words are associated more with semantics, whereas function words are linked to the syntactic aspects of the sentence (see Table 4 of Appendix C for more details about their properties). In order to investigate the relationship between predictability measures and word class in read-aloud speech, we deployed a Linear Mixed Model with speaker and POS random effect terms for duration prediction. Fixed effects included all the predictors along with interaction terms between word class and trigram lexical+PCFG syntactic surprisal measures. Each word in our dataset was annotated with a word class label (*viz.*, *content* or *function* word) derived from its universal POS tag. Table 2 (see bottom block of table) depicts the significant interaction effects between both lexical surprisal measures and word class. High values of both forward and backward trigram surprisal induced high function word duration in read aloud speech after controlling for several other factors. This result is in contrast to the asymmetric behavior observed by Bell et al. (2009) for function words in conversational English speech. See Table 6 in Appendix E for full regression model results.

Counter-intuitively, the interaction term between word class and PCFG surprisal has a negative coefficent, signifying that high values of PCFG surprisal result in low word durations for function words. Examining this anomaly, we looked at function word distributions in our dataset (TDIL corpus) and the corpus used to train the PCFG parser (HUTB corpus). Table 4 in Appendix C lists grammatical category-wise distribution of HUTB and TDIL words. Particles (3.73%) and question words (1.38%) words have higher mean surprisal and lower mean duration compared to the corresponding mean values for the function word class in TDIL corpus. The high surprisal of words belonging to these grammatical categories can be attributed to the fact that the PCFG parser training data from the HUTB corpus (particles: 1.59%, questions: 0.11%) has very few words belonging to these categories, thus impacting PCFG surprisal estimates. The following examples illustrate question words like *kis* (183ms duration and 12.16bits PCFG surprisal) and particles like *toh* (675ms and 9.5bits):

Predictor(s)	**10-fold CV prediction accuracy (%)**
Word length	68.91
+Word frequency	76.10
+Unigram surp	77.65
+Backward 3g-surp	77.02
+Forward trigram surp	77.14
+PCFG surprisal	79.61
+SC	80.21
+IC	83.94

Table 3: Prediction accuracy for content and function word classification (on the entire dataset of 15607 data points) via Generalized LMs where features are added incrementally (all differences between successive pairs of models significant at $p < 0.001$ via McNemar's test)

(1) a. yeh aag **kis** hanuman dwara lagayi
this fire WHICH hanuman by set
gayi hogi?
would?
Which Hanuman would have set this fire?

b. ab tak **toh** pitaji so gaye honge
by now PARTICLE father sleep must
By now, father must have been asleep.

The information profiles and per-word read-aloud word duration of the above examples from our dataset are presented in Figure 4 of Appendix A. Cognitively, it is also conceivable that *WH*-markers and particles might be easy to articulate being very common function words. However, they might potentially introduce complex mental operations like movement (or linking to other words in non-movement based accounts) in the upcoming structure, which are reflected in the duration of the next word (akin to spillover in reading studies). This conjecture is supported by the fact that words following question words and particles have higher duration on an average compared to the mean duration of these target function words themselves (question words: 225ms & next word 274ms; particles: 155ms & next word 292ms mean duration).

4.2.4 Word Class Prediction and PCFG Surprisal

Extending the work by Bell et al. (2009) (who do not factor in syntactic predictability estimates) described in the previous section, we explored the impact of all our measures for predicting word class using Generalized Linear Models (GLMs). For this binary classification task, function words were coded as class 1, while content words were coded as 0. Subsequently, we added each predictor incrementally to a GLM and measured the prediction accuracy of the model via 10-fold cross-validation (CV). The corpus was divided into 10 sections and 10 models trained on 9 sections each were used to generate predictions for the remaining section, thus obtaining predictions over the entire dataset. Table 3 provides CV prediction accuracies of all our incremental models. Low-level predictors, frequency and unigram surprisal, confer significant gains over a basic word length baseline. However, adding backward and forward surprisal actually worsens model performance and hence these measures do not help distinguish between content and function words. This result thus validates our findings pertaining to word class and lexical surprisal measures reported in Table 2 (bottom block). In contrast, PCFG surprisal confers a 2% increase in predicting the word class. PCFG surprisal is a more powerful measure compared to word-based surprisal models as it factors in POS tag information and syntactic context and hence outperforms word-based trigram models. DLT-costs also induce significant gains over and above models containing low-level and all other surprisal predictors. In particular, integration cost induces close to a 2.5% increase over a model containing all the other predictors.

5 Discussion

Overall, our results validate our initial hypotheses motivating the study. Linguistic complexity measures (lexical+PCFG surprisal and DLT's integration + storage costs) are significant positive predictors of word duration in reading aloud speech, mirroring trends reported in the literature on spontaneous speech production (Demberg et al., 2012; Dammalapati et al., 2021). Our measure of planning, FORWARD SURPRISAL, is also a positive predictor of reading aloud times. It potentially models parafoveal preview in the reading aspect of reading aloud. Such look-ahead during reading likely helps articulatory planning during the reading aloud process. This finding advances further support to the "*involvement-in-planning*" account, as proposed by Pluymaekers et al. (2005). The cited work shows that articulatory processes are continuous and incremental in nature; upcoming words affect the planning of the target word. Finally, our data and analyses validate the *frequency effects* (high frequency words are read aloud faster than low frequency words) predicted by the DRC model of word recognition and reading aloud.

Going further, we show that an increase in both our measures of lexical surprisal (*viz.,* backward and forward surprisal) led to identical effects on word duration of content and function words in read aloud speech, *i.e.*, increased duration for both classes of words. For the binary classification task of predicting content and function words, PCFG surprisal induces a notable boost in accuracy over a baseline containing low-level predictors and lexical surprisal measures. However, forward and backward surprisal do not help discriminate between content and function words. This is in direct contrast to the results reported by Bell et al. (2009) for spontaneous speech (Switchboard corpus). They show evidence for differential lexical access mechanisms for content and function words as attested to by the long line of work in the production literature (Garrett, 1975, 1980; Lapointe and Dell, 1989). Thus via this work, we have compared the cognitive processes in reading aloud with spontaneous speech production, an underexplored direction highlighted by Sulpizio and Kinoshita (2016) whom we cited at the outset.

Our results indicate that both content and function words might rely on the non-semantic lexical route or grapheme-phoneme correspondence (GPC) rules as hypothesized by the DRC model of reading aloud. Speakers might not be doing semantic processing during this task. The close symbol-sound correspondence in Hindi orthography (Vaid and Gupta, 2002) might be a factor contributing to this effect, a conjecture that needs to be validated using further experiments. The measure of word complexity proposed by Husain et al. (2015) and character-based surprisal models of reading difficulty proposed in recent work (Hahn et al., 2019; Oh et al., 2021) might be viable approaches towards this end. Situations where the connection between orthographic length and pronunciation length is complex (say "535" in written text articulated as *panch sau paintis*, *i.e.*, “five hundred and thirty five”) are best investigated using more controlled experimental designs.[4]

In a recent survey, Staub (2015) summarized that lexical predictability induces the graded activation of multiple upcoming words during reading comprehension (as opposed to the prediction of a single word). Moreover, lexical predictability effects occur either at the very early stages of lexical access or pre-lexical stages (processing visual features of letters in the script), rather than at post-lexical stages involving meaning identification. Based on insights from prior work, high syntactic predictability (low PCFG surprisal values in our setup) can be linked to high accessibility and hence the ease of word retrieval from memory, which in turn facilitates production ease (Bock and Warren, 1985; Arnold, 2010). Future inquiries need to tease apart the contributions of lexical and syntactic predictability in reading aloud, quantifying the impact of language-specific properties of the Hindi language on reading aloud durations. In particular, the verb-final nature of Hindi and prior findings about the interplay between expectation and locality effects (Husain et al., 2014; Ranjan et al., 2021) need to be explored. Other salient aspects like predictability and case marking (Ranjan et al., 2019), and the impact of the argument-adjunct distinction (Pandey et al., 2022), could also be investigated to contribute to a comprehensive theory of reading aloud, which accounts for data from multiple language families.

We also plan to develop reading aloud speech corpora with a larger number of participants. Moreover, the current task of reading the printed text aloud can be modified to include comprehension questions (à la reading studies) to ensure that participants engage with the material. We also plan to collect eye-tracking times to study comprehension during the reading phase prior to reading aloud. Thus this paradigm can catalyze research in integrated models of production and comprehension (MacDonald, 2013; Pickering and Garrod, 2013). Levy and Gibson (2013) point out that the surprisal measure is an *incremental* and *localized* measure of comprehension difficulty, which can be used to formalize such integrated models. Since this measure can be used to model production difficulty as well, it facilitates cross-linguistic hypothesis testing on both comprehension and production as well as interactions between these processes.

[4] We are indebted to an anonymous reviewer for this suggestion and the example.

Acknowledgements

We would like to thank Marten van Schijndel, Samvit Dammalapati, Rupesh Pandey, and members of the Cornell Computational Psycholinguistics Discussions research group (C.Psyd) for their invaluable comments and feedback on this work. We also thank IISER Bhopal undergraduate students, Chandan Upadhyay, Vipul Bhadani, Priya Kumari, Naman Joshi, and Kshitiz Dixit for their help in speech transcription. The authors acknowledge extramural funding from the Cognitive Science Research Initiative, Department of Science and Technology, Government of India (grant no. DST/CSRI/2018/263). Finally, we are also indebted to the anonymous reviewers for SCiL 2022 and AMLaP 2020 for their detailed and insightful feedback.

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).

Jennifer E. Arnold. 2010. How speakers refer: The role of accessibility. *Language and Linguistics Compass*, 4(4):187–203.

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.

Alan Bell, Jason M Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2009. Predictability effects on durations of content and function words in conversational English. *Journal of Memory and Language*, 60(1):92–111.

Alan Bell, Daniel Jurafsky, Eric Fosler-Lussier, Cynthia Girand, Michelle Gregory, and Daniel Gildea. 2003. Effects of disfluencies, predictability, and utterance position on word form variation in english conversation. *The Journal of the Acoustical Society of America*, 113(2):1001–1024.

Riyaz Ahmad Bhat. 2017. *Exploiting linguistic knowledge to address representation and sparsity issues in dependency parsing of indian languages*. Ph.D. thesis, IIIT Hyderabad India.

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.

Klinton Bicknell and Roger Levy. 2012. Why long words take longer to read: the role of uncertainty about word length. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics*, pages 21–30. Association for Computational Linguistics.

J. Kathryn Bock and Richard K Warren. 1985. Conceptual accessibility and syntactic structure in sentence formulation. *Cognition*, 21:47–67.

Max Coltheart, Kathleen Rastle, Conrad Perry, Robyn Langdon, and Johannes Ziegler. 2001. Drc: a dual route cascaded model of visual word recognition and reading aloud. *Psychological review*, 108(1):204.

Samvit Dammalapati, Rajakrishnan Rajkumar, 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.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

Vera Demberg, Asad B. Sayeed, Philip J. Gorinski, and Nikolaos Engonopoulos. 2012. Syntactic surprisal affects spoken word duration in conversational contexts. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 356–367, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kenneth I. Forster and Susan M. Chambers. 1973. Lexical access and naming time. *Journal of Verbal Learning and Verbal Behavior*, 12(6):627–635.

Lesya Y Ganushchak and Yiya Chen. 2016. Incrementality in planning of speech during speaking and reading aloud: Evidence from eye-tracking. *Frontiers in psychology*, 7:33.

Merrill Garrett. 1980. Levels of processing in sentence production. In *Language production Vol. 1: Speech and talk*, pages 177–220. Academic Press.

Merrill F Garrett. 1975. The analysis of sentence production. In *Psychology of learning and motivation*, volume 9, pages 133–177. Elsevier.

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

Michael Hahn, Frank Keller, Yonatan Bisk, and Yonatan Belinkov. 2019. Character-based surprisal as a model of reading difficulty in the presence of errors. In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019*, pages 401–407. cognitivesciencesociety.org.

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.

Samar Husain, Shravan Vasishth, and Narayanan Srinivasan. 2014. Strong expectations cancel locality effects: Evidence from Hindi. *PLOS ONE*, 9(7):1–14.

Samar Husain, Shravan 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. 2010. Redundancy and reduction: Speakers manage information density. *Cognitive Psychology*, 61(1):23–62.

Marcel A Just and Patricia A Carpenter. 1980. A theory of reading: From eye fixations to comprehension. *Psychological review*, 87(4):329.

Steven G Lapointe and Gary S Dell. 1989. A synthesis of some recent work in sentence production. In *Linguistic structure in language processing*, pages 107–156. Springer.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126 – 1177.

Roger Levy and Edward Gibson. 2013. Surprisal, the pdc, and the primary locus of processing difficulty in relative clauses. *Frontiers in Psychology*, 4(229).

Maryellen C. MacDonald. 2013. How language production shapes language form and comprehension. *Frontiers in Psychology*, 4(226):1–16. Published with commentaries in Frontiers.

John Morton. 1969. Interaction of information in word recognition. *Psychological review*, 76(2):165.

Eduardo Navarrete, Bradford Z Mahon, Anna Lorenzoni, and Francesca Peressotti. 2016. What can written-words tell us about lexical retrieval in speech production? *Frontiers in psychology*, 6:1982.

Byung-Doh Oh, Christian Clark, and William Schuler. 2021. Surprisal estimators for human reading times need character models. 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 3746–3757, Online. Association for Computational Linguistics.

Rupesh Pandey, Sidharth Ranjan, and Rajakrishnan Rajkumar. 2022. Locality effects in the processing of argument structure and information status using reading aloud paradigm. In *Proceedings of the 8th Annual conference of the Association for Cognitive Science (ACCS)*, India. Amrita University.

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.

Martin J. Pickering and Simon Garrod. 2013. An integrated theory of language production and comprehension. *Behavioral and Brain Sciences*, 36:329–347.

José C Pinheiro and Douglas M Bates. 2000. Linear mixed-effects models: basic concepts and examples. *Mixed-effects models in S and S-Plus*, pages 3–56.

Mark Pluymaekers, Mirjam Ernestus, and R Harald Baayen. 2005. Lexical frequency and acoustic reduction in spoken dutch. *The Journal of the Acoustical Society of America*, 118(4):2561–2569.

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. 2020. Forward surprisal models production planning in reading aloud. In *Proceedings of the 26th Architectures and Mechanisms for Language Processing Conference (AMLaP)*, Potsdam, Germany. University of Potsdam.

Sidharth Ranjan, Rajakrishnan Rajkumar, and Sumeet Agarwal. 2021. Locality and Expectation Effects in Hindi Preverbal Constituent Ordering. *Cognition*, in press.

Keith Rayner, Sara C Sereno, and Gary E Raney. 1996. Eye movement control in reading: a comparison of two types of models. *Journal of Experimental Psychology: Human Perception and Performance*, 22(5):1188.

Elizabeth R Schotter, Bernhard Angele, and Keith Rayner. 2012. Parafoveal processing in reading. *Attention, Perception, & Psychophysics*, 74(1):5–35.

Gregory Scontras, William Badecker, Lisa Shank, Eunice Lim, and Evelina Fedorenko. 2014. 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):1503–1514.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Adrian Staub. 2015. The effect of lexical predictability on eye movements in reading: Critical review and theoretical interpretation. *Language and Linguistics Compass*, 9(8):311–327.

Andreas Stolcke. 2002. SRILM — An extensible language modeling toolkit. In *Proc. ICSLP-02*.

Simone Sulpizio and Sachiko Kinoshita. 2016. bridging reading aloud and speech production. *Frontiers in psychology*, 7:661.

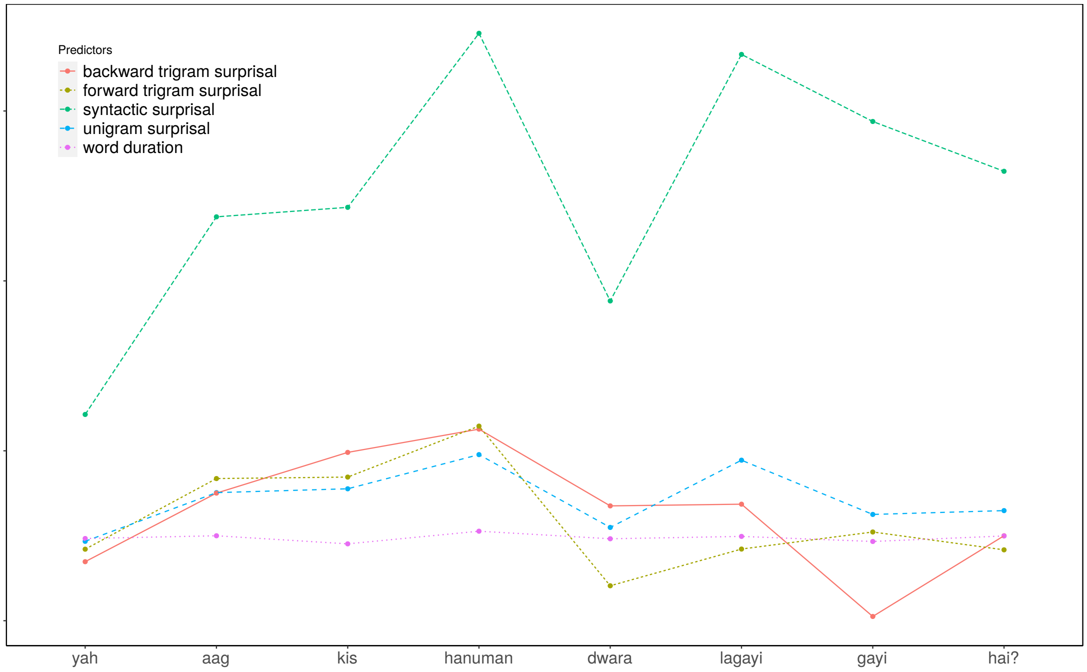
J Vaid and Anshum Gupta. 2002. Exploring word recognition in a semi-alphabetic script: The case of devanagari. *Brain and Language*, 81:679–90.

Marten van Schijndel, Andy Exley, and William Schuler. 2013. A model of language processing as hierarchic sequential prediction. *Topics in Cognitive Science*, 5(3):522–540.

Himanshu Yadav, Ashwini Vaidya, and Samar Husain. 2017. Keeping it simple: Generating phrase structure trees from a Hindi dependency treebank. In *TLT*.

A Information Profile

Figure 4 depicts the information profiles of Examples 1a and 1b respectively from the TDIL corpus discussed in Section 4.2.3 of the paper.



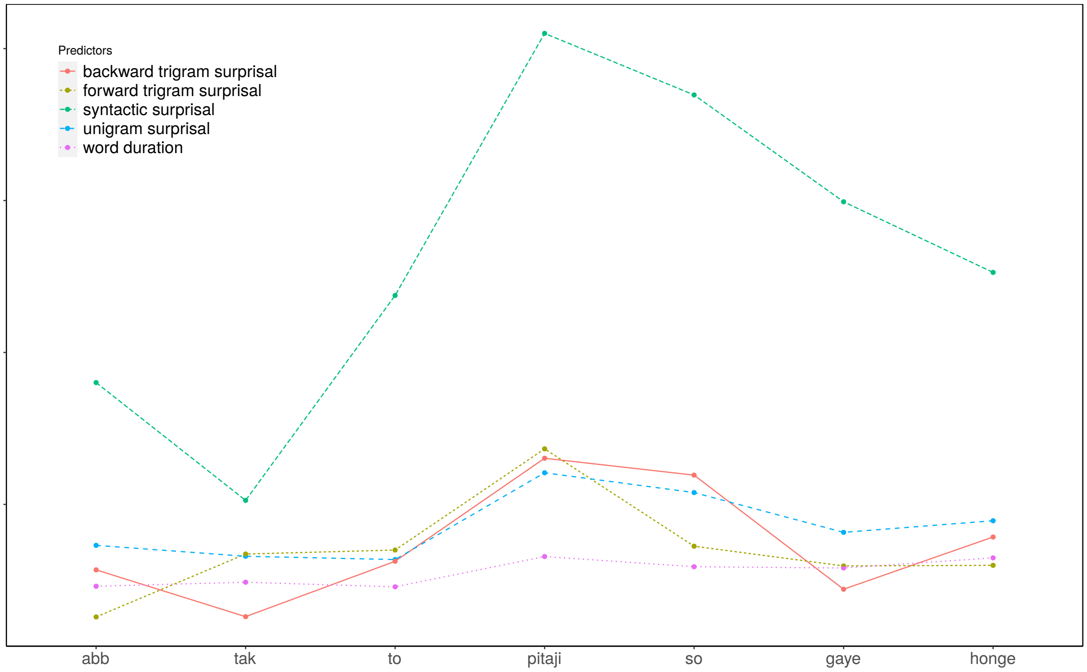


Figure 4: Word duration and information profiles of sentences containing a question marker (*kis*; top figure) and particle (*toh*; bottom figure)

B Dual Route Cascaded (DRC) Model

The DRC model is shown in Figure 5. Each route consists of several interacting layers containing a set of units (representing words in the orthographic lexicon or letters in the letters layer). Units of different layers interact via *inhibition* (an activated unit impedes activation levels of other units) or *excitation* (an activated unit facilitates activation of other units). Figure 3 shows a snapshot of parafoveal preview in reading.

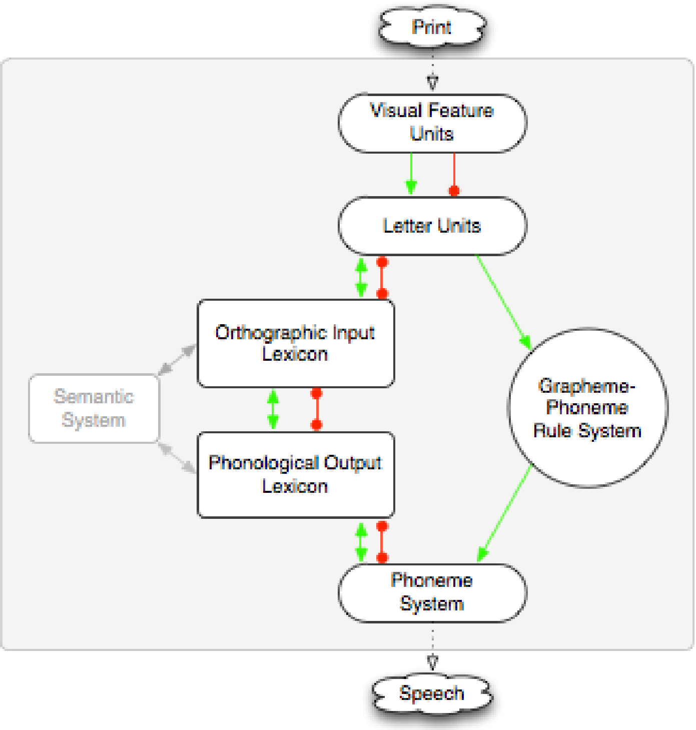

Print
Visual Feature Units
Letter Units
Orthographic Input Lexicon
Semantic System
Grapheme-Phoneme Rule System
Phonological Output Lexicon
Phoneme System
Speech

Figure 5: DRC model[a] of visual word recognition and reading aloud by Coltheart et al. (2001)

[a] Reproduced from: https://maxcoltheart.wordpress.com/drc/

C Details of Hindi Script and Grammatical Categories

Unlike the Latin alphabet, Hindi has no concept of letter case (upper/lower) except for sinistrodextral (left-to-write) writing system. Each unit of word is written in horizontal direction separated by space and follows standard punctuation markers alike English except for full stop (.) where a pipe (।) is used as an end of sentence marker. Vowel diacritics (glyph) combines with consonants to form another syllabic letter (आ + क = का). For example, the vowel – आ ($\bar{a}$) combines with consonant – क (k) to give a letter का ($k\bar{a}$) with added vowel sign in diacritic form. Conjunct consonants is understood to offer most difficulty during reading consist of two consonants grouped together but with a missing vowel sound between them. For example, the two consonants (च, छ) when combined together (च + छ = च्छ), the letter च्छ (as in the word– अच्छा) has a missing vowel (अ) diacritic i.e., ा between them.

Table 4 illustrates the distribution of various grammatical categories in TDIL and HUTB corpora of Hindi written text as well as properties of content and function words. The mean word length of a content word in the TDIL corpus was 2.66 (minimum: 1, maximum: 8), and the function word was 1.74 (minimum: 1, maximum: 5).

Category	**%Freq**	**%Freq**	**Length**	**PCFG**	**RT**
■	273013 words	15607 words	characters	**surprisal**	ms
Corpus	HUTB	TDIL	Mean values in TDIL		
CONTENT					
Verb	18.12	32.15	1.98	11.26	274.99
Noun	38.47	26.96	2.86	13.92	375.45
Adjective	5.91	3.71	3.01	14.53	399.65
Adverb	0.47	0.78	2.88	14.21	367.91
FUNCTION					
Postposition	21.42	11.14	1.22	5.58	178.22
Pronoun	4.34	11.07	2.20	10.62	258.12
Det	4.65	4.64	1.86	8.87	242.32
Particle	1.59	3.73	1.16	**8.42**	**155.02**
Conjunction	4.13	3.17	1.87	7.65	206.01
Question	0.11	1.38	0.11	**12.59**	**225.97**
Quantifier	0.81	1.27	0.81	11.23	294.96
Content words	62.97	63.70	2.66	13.96	354.29
Function words	37.03	36.40	1.74	8.60	226.54
All words	100.00	100.00	2.17	11.10	286.17

Table 4: Grammatical category-wise descriptive statistics in TDIL and HUTB corpora

D PCFG Parser Training Procedures

Following steps were involved in training the Modelblocks parser using the HUTB corpus:

1. The parser training requires phrase-structure trees as input. Due to the unavailability of such resources in Hindi, we created our own corpus by converting the existing dependency parsed trees (Dependency structure; DS) of HUTB corpus (Bhatt et al., 2009) into constituency parsed trees (Phrase structure; PS) using an approach described in Yadav et al. (2017).

2. However, we had to do some extra post-processing of the obtained phrase structure trees (removal null nodes, unary nodes, punctation and coordination fixes, inter-alia) to make it compatible with the format expected by the Berkeley parser. The corrected final phrase structures thus produced were used to train the Berkeley parser model.

3. Parser training involved estimating a sophisticated grammar using 4 iterations of the split-merge algorithm (Petrov et al., 2006) and a beamwidth of 5000 (shown to be effective for reading time studies).

E Interaction analysis of word class and word length with surprisal

Predictors	**Estimate**	**Std. Error**	**t-value**
Intercept	5.550	0.098	56.825
Word length	0.237	0.004	60.946
Unigram surprisal	0.039	0.006	6.118
Word frequency	-0.004	0.005	-0.777
IC	-0.018	0.003	-6.550
SC	0.005	0.004	1.106
Backward surprisal	0.028	0.005	5.556
Forward surprisal	0.044	0.005	9.653
PCFG surprisal	0.034	0.005	6.904
INTERACTIONS			
Word length x Backward 3g-surp	-0.024	0.004	-5.491
Word length x Forward 3g-surp	-0.031	0.004	-8.061
Word length x PCFG surprisal	0.001	0.005	0.314

Table 5: Fixed effects of LMM (with word length as interaction term) predicting reading aloud time (15607 data points; all significant predictors denoted by |t|>2)

Predictors	**Estimate**	**Std. Error**	**t-value**
Intercept	5.512	0.099	55.652
Word length	0.216	0.003	62.147
Unigram surprisal	0.036	0.007	5.451
Word frequency	-0.028	0.005	-6.038
SC	0.012	0.005	2.517
IC	-0.016	0.003	-5.171
Backward 3g-surp	0.007	0.006	1.095
Forward 3g-surp	0.018	0.005	3.364
PCFG surprisal	0.065	0.007	9.192
Word class	0.024	0.011	2.264
INTERACTIONS			
Function word x Backward 3g-surp	0.028	0.009	2.936
Function word x Forward 3g-surp	0.041	0.008	4.855
Function word x PCFG surprisal	-0.039	0.009	-3.953

Table 6: Fixed effects of LMM (with word class as interaction term) predicting reading aloud time (15607 data points; all significant predictors denoted by |t|>2)

Modeling human-like morphological prediction

Eric Rosen
University of Leipzig
errosen@mail.ubc.ca

Abstract

We test a model of morphological prediction based on analogical deduction using phonemic similarity by applying it to German plural suffix prediction for a set of 24 nonce forms for which McCurdy et al. (2020) elicited human judgements, and which they found were poorly matched by productions of an encoder-decoder model of Kirov and Cotterell (2018). Their results raise the question of what kinds of models best mirror human judgements. We show that the predictions of the analogical models we tested mirror human judgements better than the encoder-decoder model.

1 Do neural models of morphological prediction emulate human behaviour?

Despite the recent success of neural models of morphological prediction such as the encoder-decoder (ED) model of Kirov and Cotterell (2018) (henceforth KC), two recent papers: Corkery et al. (2019) and McCurdy et al. (2020) (henceforth CMG and MGL) question how well these models' predictions of nonce forms match those of human judgements. Corkery et al. (2019) reexamine KC's application of their ED model to English past-tense nonce forms developed by Albright and Hayes (2003) (henceforth AH) through multiple random initializations of their model and find that KC's model predictions do not align with AH's results as well as reported by KC.

MGL pursue this question further by eliciting human judgements of possible German plural forms of 24 nonce words originally developed by Marcus et al. (1995) (henceforth M95): 12 'rhymes' with regular phonotactic patterns and 12 phonologically atypical 'non-rhymes', shown in table 1. As MGL put it, KC's claim, that "modern Encoder-Decoder (ED) architectures learn human-like behavior when inflecting English verbs, such as extending the regular past tense form to novel words" does not address a point made by M95: that neural models "may learn to extend not the regular, but the most frequent class – and thus fail on tasks like German number inflection, where infrequent suffixes like /s/ can still be productively generalized." As did CMG with AH's English nonce forms, MGL apply KC's ED model to M95's German nonce forms and compare them with their elicited human judgements. They find that the ED model fails to match human prediction in German plural formation, where, unlike in English, no class holds a majority.

Outline of the paper Here, we test to what extent an alternative model that predicts forms through analogical implicative relations can improve on an ED model for matching human prediction. In the rest of §1, we further discuss how MGL's wug test results compare with those of the ED model. In §2 we present variations on an alternative model of nonce word prediction. In §3 we compare the predictions of our model with MGL's human predictions. In §4 we compare our model with other models. In §5 we report tests made on real data. §6 concludes with a discussion.

Rhymes	Non-Rhymes
pind	fnahf
kach	pläk
spand	pnähf
spert	plaupf
klot	pröng
bral	fnöhk
raun	fneik
mur	bnöhk
vag	snauk
nuhl	pleik
pund	bnaupf
pisch	bneik

Table 1: 24 nonce forms developed by M95 and tested by MGL

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 133-142.
Held on-line February 7-9, 2022

MGL's wug test results Table 2, reproduced from MGL, shows MGL's wug-test results in percentages for each suffix. They find a high degree of variability among speaker data, where no plural class dominates, and /e/ is the most common suffix at around 45%. /en/ and /s/ are more common in non-rhymes than in rhymes. /er/ is less common in non-rhymes. Relatively low ratings for /s/ conflict with M95 who claim that /s/ is a default suffix that can apply in any environment.

Plural		Prod %
/-e/	R	45.3
	NR	44.4
/-(e)n/	R	25.0
	NR	34.7
/-er/	R	17.4
	NR	6.7
/-s/	R	4.2
	NR	6.4
/-∅/	R	2.7
	NR	2.7
other	R	5.4
	NR	4.8

Table 2: MGL's survey results (R=rhymes, NR=non-rhymes

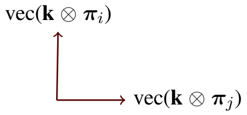
The coloured bar graphs in figure 1 (p. 5), reproduced from MGL, illustrate the differences in suffix prediction between the speaker data and their test of KC's ED model on the same nonce forms. The graphs show that the ED model predicts /en/ (purple) on the nonce forms way less than speakers. MGL suggest that the ED model over-predicts /e/ (blue) because of its frequency and does not capture minor patterns. They also observe that speaker production of /(e)n/ (purple) and /s/ (orange) is greater for Non-Rhymes relative to Rhymes. In the ED model, the tendency is reversed, where /e/ occurs for over 90% of Non-Rhymes.

2 An alternative model

As an alternative to an ED model, we explore morphological prediction through *implicational relations* (Bonami and Beniamine, 2016; Ackerman and Malouf, 2016; Ackerman et al., 2009b,a) based on phonological similarity. Because we are trying to predict a plural from an affixless singular, we can't use principal parts and we can only guess an inflectional class through phonological clues and possibly what the phonology might suggest about semantics. If a speaker knows both the singular and plural of lexeme A, they can predict the plural of lexeme B from the singular if lexeme B is similar to A and forms the plural in the same way. e.g.: Fisch → Fische ('fish(es)'), Tisch → Tische ('table(s)')

We adopt a Vector Symbolic Architecture (VSA) model (Kanerva, 2009, 1988, 2017)[1] for representing sequences of phonemes, in which vectors are binary, with a typical dimension of 10,000. A phonological feature is represented by a randomly chosen sparse binary vector. The vector for each feature will be nearly orthogonal to all other features' vectors. A phoneme is represented by the sum of the feature vectors that compose it: for example, **k** = **cons** + **dorsal**, with features **sonor**, **voi**, **cont** at zero. (Bolded terms are vectors.) **g** differs from **k** just by the addition of feature **voi**. Each phoneme needs no more than 7 features to be represented. Basing phonemes on features means that the vectors of phonologically similar segments in the same position will be relatively close in the space (e.g., /k/ and /g/), if they differ by just one feature and relatively far (e.g., /k/ and /o/) if their features are mostly different.

To represent a *sequence* of phonemes, we superpose the encodings of all the phonemes, but each phoneme vector is cyclically permuted by one bit for each step in the sequence. Permutation moves a vector to a part of the space where it is nearly orthogonal to its non-permuted position and thus to where it will not interfere with other vectors as shown below. In this framework we can use phonological features in order to make deductions based on feature similarity.


vec($\mathbf{k} \otimes \boldsymbol{\pi}_i$)
vec($\mathbf{k} \otimes \boldsymbol{\pi}_j$)

Implicative relations (Ackerman and Malouf 2016, inter alia): e.g., *Bratsche* : *Bratschen* ::

[1] As noted by an anonymous reviewer, nothing in the analysis hinges on the particular model we are using for representing sequences of phonemes. We adopt the VSA model here for convenience, but what is crucial is the idea of predicting by feature-based similarity and (to be discussed below) word frequency.

Patsche : *Patschen* ('viola' sg : pl :: 'paw' sg : pl) are predicted by vector differences where $y_{pl} \simeq y_{sg} + x_{pl} - x_{sg}$ for lexemes x and y whose phonological-feature-based vector encodings are similar according to some similarity metric. Unlike conventional neural models, our model has no network and requires no training. Although the scores for choosing predictors have continuous values, the vector representations are effectively discrete.[2]

Nouns from the Unimorph dataset are used in conjunction with two frequency archives: Institut für Deutsche Sprache (2014) and Gambolputty. We convert both singular and plural forms to a phonemic representation using the German version of Bernard and eliminate a handful of words given non-German phonemes such as *psychothriller* (θ) or *chance* (ã) to end up with 36 phonemes, encodable with 16 phonological features.

Encoding German words To predict an unknown plural form[3] of lexeme A from its singular, we look for a lexeme B whose plural form is known and whose representation of the singular is close to lexeme A's. For example, *Kind* 'child' is a possible candidate for predicting the plural of nonce *pind*. If the two singular forms being compared are unequal in length, we pad the left edge of the shorter one with dummy phonemes represented by zero vectors so that their right edges align.

Calculating the score of a predicted suffix for a given word We explored different possible combinations of hyperparameters for the model to see how well the results of each marched MGL's human predictions. The hyperparameters included the following, where the hyperparameter choice for the results given below is starred:

1. The similarity metric for choosing predictive best neighbours of a nonce form. Calculating on raw cosine similarity between the vector for the nonce word and a candidate word did not spread out the values enough to sufficiently distinguish similar words from dissimilar ones.
   - Reciprocal of sum squared vector difference.
   - Further squaring the above value.
   - Reciprocal of sum of the absolute values of the difference of vectors.
   - *log $\frac{1}{1-s}$, where s is the cosine similarity of the vectors.

2. The frequency score for each candidate word. We tried:
   - *Raw frequency.
   - Log frequency.
   - Squared raw frequency to spread the values out more and penalize infrequent words more as candidates.

3. The width of a beam search (*beam=6) among top-scoring neighbour candidates.

4. *Comparing the best candidate(s) for each possible suffix rather than just the suffixes that appear among the top candidates.[4]

5. *Scaling the similarity score to increase towards the end of the word. When taking the cosine distance between the vectors of a nonce word and a neighbour we take not the raw vectors but vectors where the values of the component for each phoneme in the string are boosted by factor s^i, where s is a scaling factor such as 1.2 and i is the ordinal position of the phoneme in the string. e.g., for nonce word *spand*, *Pfand* 'pledge, deposit' and *Brand* 'fire' would be better predictors than *Spalt* 'crack' or *Spatz* 'sparrow'.

6. *Weighting the score by the negative exponential of the syllable count difference between the candidate and the source word so that analogies are biased to be based on prosodically similar words. (e.g., quadrisyllabic *Geburtstagskind* 'birthday child' is scored lower than *Kind* as a predictor for nonce *pind*.)

7. *Adding a score for each suffix based on the probability of each suffix in the candidate nonce word as assigned by a single-layer

[2] MGL trained the ED model on nouns in orthographic form and say "Unlike English, the phonological-orthographic mapping is straightforward in German, so we can use a written corpus for model training." This isn't quite true, given the non-negligible occurrence of foreign words in the corpus like *Babysitter, Boutique or Clique*, whose German pronunciations are idiosyncratic or mutually inconsistent.

[3] MGL abstract away from questions of umlaut. See Trommer (2021) for a detailed analysis of the interaction between gender, plural allomorphy and umlaut.

[4] The former option was suggested by Matías Guzmán Naranjo (p.c.) and yielded better results.

RNN trained on all the plurals of words in the database with frequencies above 100,000.

As noted by an anonymous reviewer, it is possible to engineer the choice of the above hyperparameters to make the results match MGL's human predictions as closely as possible. If it is the case that the mode of human prediction of nonce forms closely mirrors human prediction of real data, then these engineered choices are overfitting to the extent that they diverge from a model that is trained on and best predicts real data. On the other hand, it may not be the case that human speakers predict nonce words the same way they predict real words. The difficulty in getting a model to work equally well on prediction of real words and on nonce forms is noted by CMG (p. 3874 §5.1), who write: "It seems that the ED model displays a fundamental tension between correctly modelling humans on real words and nonce words." (p. 3874) A possible reason for this tension is given by Schmitz et al. (2021), who propose that nonce words are not "semantically empty shells" and that "[t]he resonance of morphologically simplex and complex pseudowords with the words in the mental lexicon influences the processing of these pseudowords." (slide 8) If their hypothesis is correct, then speakers judging these 24 nonce words may be using associations between these words and real words that are based not on the kinds of phonological similarities that our model measures, but instead on the kinds of onomatopoeic or phonaesthematic associations that Schmitz et al. (2021) suggest. In fact, we find that the hyperparameter choice that best predicts suffixes of real data does not necessarily best mirror MGL's human prediction results. As an illustration of the mismatch between nonce word and real data prediction, figure 3 graphs the nonce word predictions made by the same model that performed best (85% accuracy) on real data. This model over-predicts that /-s/, null and in some cases the 'other' and /-er/ suffixes. It should be understood then, that the results shown below illustrate how an ideal choice of hyperparameters can mirror MGL's nonce word predictions but they should not be considered as a held-out test set of real-data training.

Table 3 shows the normalized scores and top candidate for each suffix for the first nonce word *pind* under one hyperparameter combination.

Suffix	Best neighbour	Gloss	Score
er	Kind	'child'	0.474
e	Wind	'wind'	0.392
en	Mensch	'human'	0.081
s	Trend	'trend'	0.025
null	Cent	'cent'	0.020
oth	Konzern[5]	'corporation'	0.004

Table 3: Top neighbour and normalized score for each suffix for MGL's first nonce word *pind*.

3 Graphical inter-model comparisons

Figure 1 compares predictions of MGL's human subjects with the ED model and Fig. 2 with one variation of the implicational model. The ED model greatly over-predicts the /e/ suffix at the expense of the other suffixes. The implicational model does not exactly match MGL's human predictions but we can see some patterns in common. A strong score of the /er/ suffix (green) in the first two nonce words occurs in both, but the implicational model is weaker on /er/ than human prediction on the third and fourth nonce words. The /e/ suffix (blue) is strong in the last two rhymes in both but is slightly weaker overall in the implicational model than in MGL's human judgements.

The /s/ suffix (orange) is somewhat over-predicted by the implicational model but mirrors the human judgements with a stronger overall prediction in non-rhymes than in rhymes. Overall over-prediction of /s/ by the implicational model is likely due to an abundance of foreign borrowings with this plural form in the Unimorph dataset. This can be seen if we calculate the frequencies of suffixes in the dataset and compare with the figures given by MGL, as shown in table 4. These calculations, using the frequency score from Institut für Deutsche Sprache (2014), are fairly close to the numbers given by MGL but /s/ is slightly higher.

[5] This word was incorrectly given a suffix [ee] instead of [ə] by the phonemizer.

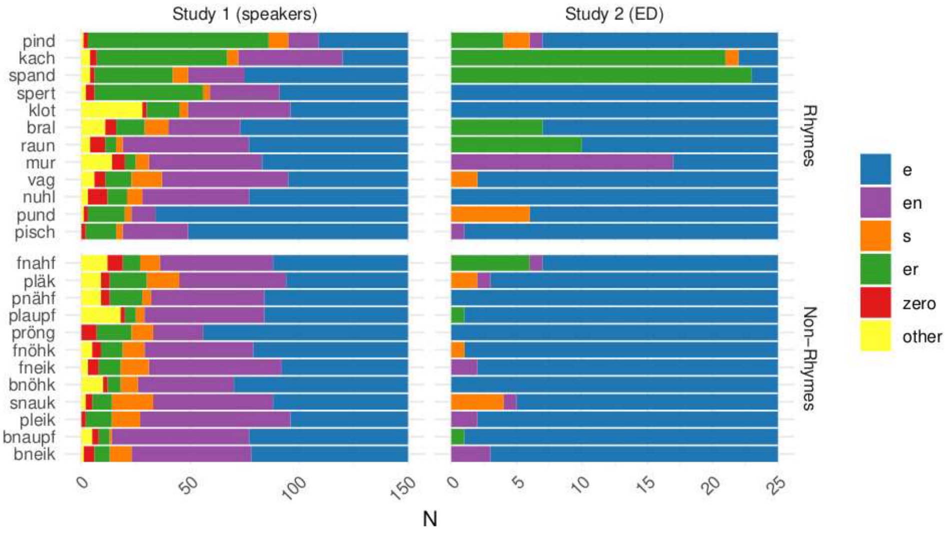


Figure 1: Plural class productions by item.

Figure 1: MGL's plural class productions compared to those of the ED model

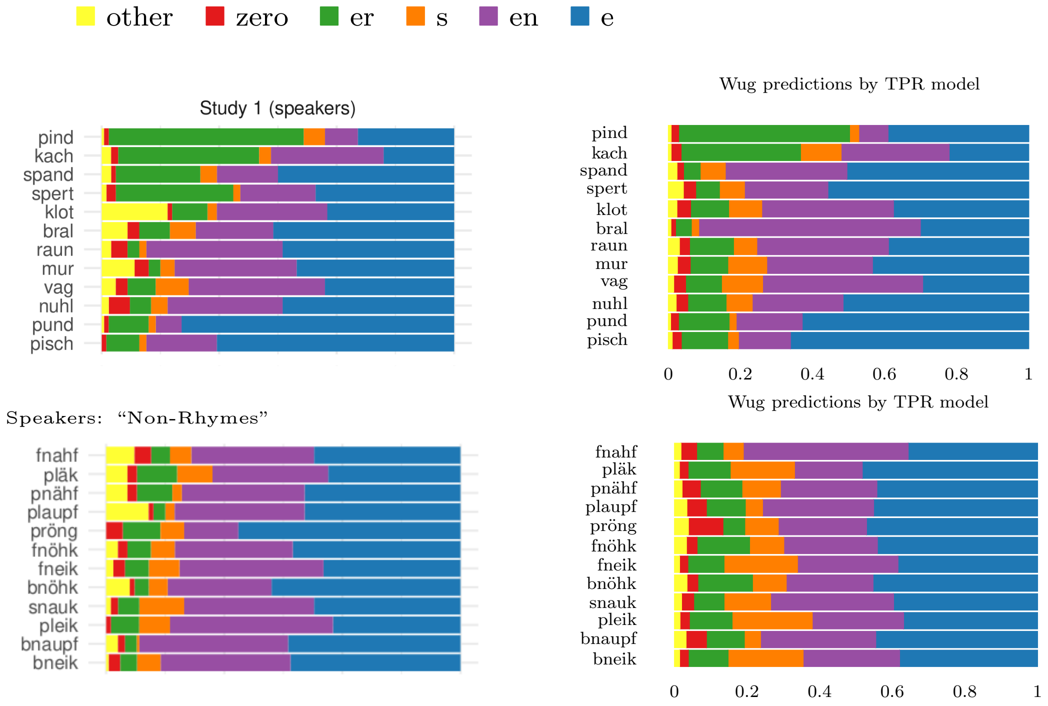


Figure 2: MGL's plural class productions compared to those of the implicational model

Suffix	Type (MGL)	Token (MGL)	Calculated here
/-(e)n/	.48	.45	.450
/-e/	.27	.21	.265
/-∅/	.17	.29	.189
/-er/	.04	.03	.035
/-s/	.04	.02	.048
other	–	–	.013

Table 4: Frequencies of suffixes in MGL's results and those produced by the current model

Calculating Spearman rank correlations MGL calculate the Spearman rank correlations between ED model production ranks and those of human speakers for each suffix across all nonce forms. They conclude from their results that there is no "statistically significant difference from the null hypothesis of no correlation." Following their approach, we perform a similar calculation to compare one set of implicational model results with MGL's speaker judgements. Table 5 shows the rank of each suffix for each nonce word for the implicational model's predictions and MGL's speaker judgements (IMP:MGL).

Suffix Nonce	oth	∅	er	s	en	e
pind	6:6	5:5	1:1	4:4	3:3	2:2
kach	6:5	5:6	1:1	4:4	2:2	3:3
spand	5:5	6:6	4:2	3:4	2:3	1:1
spert	5:6	6:4	4:2	3:5	2:3	1:1
klot	6:3	5:6	3:4	2:5	2:2	1:1
bral	6:5	5:6	3:3	4:4	1:2	2:1
raun	5:5	6:3	3:4	4:6	2:2	1:1
mur	6:3	5:5	4:6	3:4	2:2	1:1
vag	6:5	5:6	4:4	3:3	1:1	2:2
nuhl	6:6	5:4	3:3	4:5	2:2	1:1
pind	6:6	4:5	3:2	5:4	2:3	1:1
pisch	6:6	5:5	3:3	4:4	2:2	1:1
fnahf	6:3	5:6	3:5	4:4	1:2	2:1
pläk	6:5	5:6	4:3	3:4	2:2	1:1
pnähf	6:4	5:6	3:3	4:5	2:2	1:1
plaupf	6:3	4:6	3:4	5:5	2:2	1:1
pröng	5:6	4:5	6:3	3:4	2:2	1:1
fnöhk	6:5	5:6	3:3	4:4	2:2	1:1
fneik	5:6	6:5	4:4	3:3	2:1	1:2
bnöhk	5:3	6:6	3:5	4:4	2:2	1:1
snauk	6:6	5:5	4:4	3:3	2:2	1:1
pleik	6:6	5:5	4:4	3:3	2:1	1:2
bnaupf	6:3	4:5	3:4	5:6	2:2	1:1
bneik	6:6	5:5	4:4	3:3	2:2	1:1
$\sum d^s$	59	28	37	19	7	4

Table 5: Rank comparisons: speaker judgements and implicational model

Calculating the Spearman rank correlation between MGL's speaker judgements and model productions for each suffix, we get the correlations shown in table 6 for those calculated by MGL (ρ_{ED}) and for the implicational productions (ρ_{IMP}). The results show that the relative ranks for each suffix for each nonce word mirror those of MGL's wug tests fairly closely.

Suffix	ρ_{ED}	ρ_{IMP}	p_{IMP}
oth	n.a.	0.972	< 0.001
∅	n.a.	0.987	< 0.001
er	0.05	0.983	< 0.001
s	0.33	0.991	< 0.001
en	0.28	0.997	< 0.001
e	0.13	0.998	< 0.001

Table 6: Rank correlations for each suffix

Pearson correlations Table 7 shows the calculated Pearson correlation between MGL's production scores and the implicational model's for each of the six suffixes. Calculated individually, 3 of the 6 suffixes show significant correlation. But calculated across all suffixes we see strong correlation.

Suffix	r	p-value	significant
oth	0.159	.458	no
∅	0.318	.096	no
er	0.748	.00001	yes
s	0.578	.003	yes
en	0.340	.103	no
e	0.713	.0001	yes
all suffs.	0.902	$\simeq 0$	yes

Table 7: Pearson correlations

A possible reason for the model's poorer correlation for suffix ∅ is corpus noise. For the first six nonce words the highest scoring predictive words are dominated by spurious referents: (a) non-nouns such as *zweifel* 'second' or *drittel* 'third' with null plurals, (b) words given an incorrect null plural such as *cent* 'cent', or (c) proper names like *Siemens* or *Lutz*. And /en/'s poorer correlation is due to nonce words that got low scores for /en/ in MGL's wug tests in which our model over-predicts /en/ as a result of measuring similarity by featural closeness rather than an exact structural description. For example, *spand* gets a low score for /en/ from speakers but a high score from top candidate with an /en/ plural *Mensch* 'human', whose vowel and final consonant differ minimally from those of *spand* in their features.

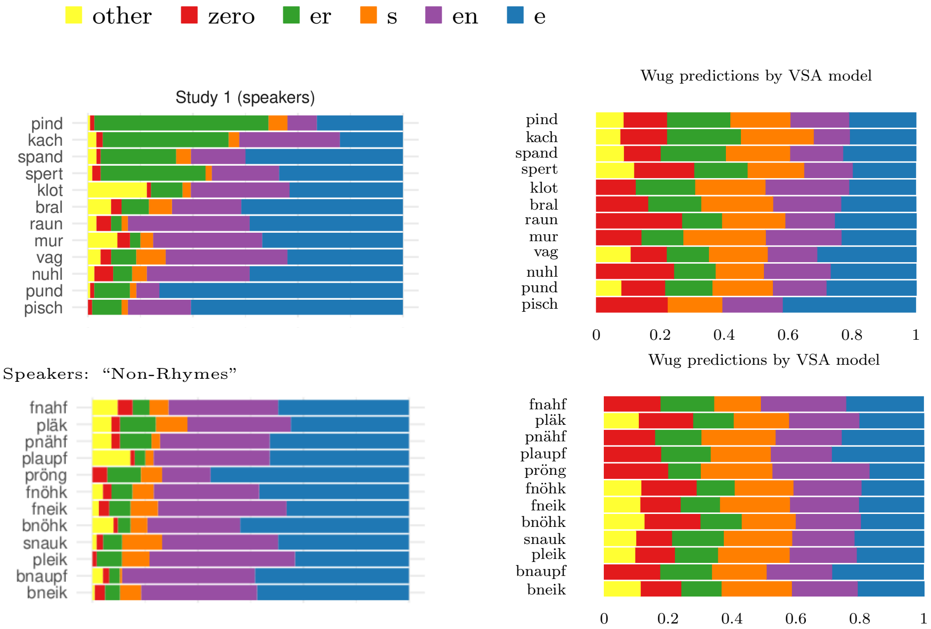


Figure 3: MGL's plural class productions compared to those of a variation implicational model that worked well on real data

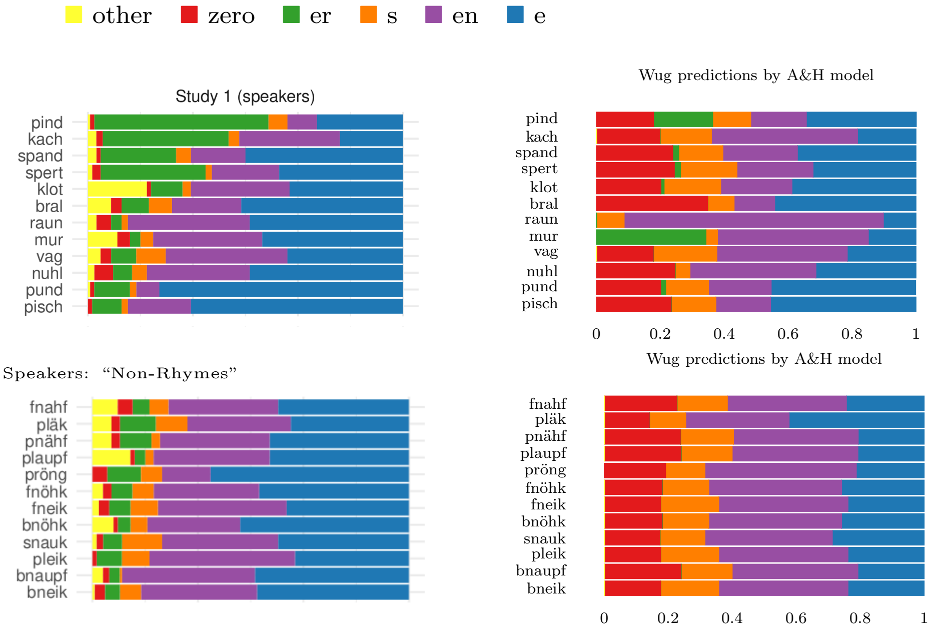


Figure 4: MGL's plural class productions compared to those of Albright and Hayes' rule-based model

In summary, our Spearman and Pearson correlation results indicate that the implicational model aligns moderately well with the rankings of suffixes for each nonce word but doesn't always reflect the differences in suffix preferences by speakers among the nonce words – especially for the suffixes 'oth', 'null' and /en/.

4 Other models

Testing with Albright and Hayes' rule-based model We also ran the nonce words through the rule-based model of Albright and Hayes (2003) trained on the Unimorph corpus. Graphical results are shown in figure 4. We found that the model over-predicts the null suffix at the expense of the /er/ suffix, which hardly occurs at all. A possible reason is that the rule-based model requires an exact structural description to trigger a rule. The null suffix occurs abundantly, e.g., in nonce word *pind* because of a default rule with strength 0.396 that makes no changes to a stem ending in one of {d, l, n, r, s, t, z} if no other rules are triggered. For the /er/ suffix to occur requires a very specific rule or else a default rule with strength 0.001.

Other rule-based and symbolic models Further testing with other rule-based models could determine how well rule-based models can model MGL's human wug prediction. Payne et al. (2021) test a rule-based model based on Yang (2016)'s Tolerance Principle using morphosemantic and phonological features that include gender features when tested on German plural formation. They test stochastically sampled nouns from German CELEX, so it remains to be tested what their model would predict for MGL's wug forms.

Beniamine and Naranjo (2021) take an approach to morphological prediction that shares some common elements with ours. They use multiple alignments of forms in inflectional paradigms. Versions of our model that do not truncate a candidate word to equalize its length with the nonce word do predict stem changes such as umlaut[6], but because we are comparing with MGL's results, which abstract away from stem changes, we do not allow for possible gaps in alignment. On the other hand, because our VSA model uses binary vectors in a distributed representation, graded, continuous degrees of similarity can be used in a way that is not possible with purely symbolic models.

Calderone et al. (2021) report on their morphological prediction experiments on nonce verb forms in English, German and Dutch, using several variations of a model that combines a bidirectional LSTM with 'fine alternation patterns' that figure in analogical deduction of word forms. They report Pearson correlations for regular and irregular verbs for their best-performing model along with Albright and Hayes' Minimal Generalization Learner and a purely analogical model of Nosofsky (1990). It is difficult to compare their results with ours because they are dealing with verb inflection rather than noun plurals and systems that have a clear regular/irregular split. They report ratings of 0.583 and 0.595 for regulars and irregulars respectively, which roughly compares with our results for the /s/ suffix and are lower than our result for /e/ and /er/. The results of their model, which, like ours, uses analogical deduction, but in a different way, provides further support for the role of analogical deduction in morphological prediction.

One approach that we did not take was to present all the nonce forms as neuter as MGL appear to have done. Among the 17,488 neuter nouns in the dataset, only 83, or 0.48% have an /(e)n/ suffix. Given the relatively strong presence of this suffix in MGL's wug predictions, it is not clear how presenting each nonce form as neuter would produce such results.

5 Testing on real data

The Unimorph dataset As mentioned above, we found that there was an inconsistency between model variations that best predicted the suffixes of real words in the Unimorph dataset and those that best matched MGL's results for nonce word prediction. We found that using log frequencies rather than raw frequencies gave better results for real-word prediction, so that infrequent words would have more weight, since many infrequent words in the dataset that we are testing will also have infrequent phonological neighbours. For real-word prediction, we also did not use RNN-generated perplexities which were specifically tailored to the wug words and whose main purpose was to allow suffixes that were a not a first choice for a wug word to have non-zero scores. The model achieves 85.6% accuracy on a sample of 3,390 items as compared with 88.8% reported by MGL with the ED model. The ED model arguably has an advantage in identifying foreign words in that it used

[6] For example, some predictions for nonce word *kach* with the /-er/ suffix produce /kɛçɛr/ with umlauted /a/.

orthographic rather than phonemic input, which gives clues to a word's foreignness. For example a word spelled with c followed by a letter other than h or k is likely foreign. Foreign words make up a sizeable portion of words our model misses: for example *mustang*, *body*, *kanu* 'canoe', *strip*, *gun*, *overtime*.

6 Discussion

As discussed above, we tested many variations of the model, for which there is not space to list the details of each one's result. The variation that made the most difference to the results was the inclusion of frequency scores. A further step with this model is introduce learning, so that instead of having positional vectors intentionally orthogonal, they are allowed to move together closer in the space so that a phoneme in one position can have some measured similarity with the same phoneme in a nearby position.

Given that right-aligned edge calculation, features of segments and prosodic shape of implicational candidates were all found to contribute to predicting plurals of nonce forms based on word similarity, it is notable, as observed by an anonymous reviewer, that word similarity appears to be a multidimensional calculation that involves all of these properties.

The fact that MGL's wug tests results give non-negligible scores to all the suffix classes for most of the 24 items suggests that each suffix has found some niche or set of niches in the sense of Aronoff (2021), who gives the example of the ongoing niche competition between English /-er/ $\sim$ /-est/ and adverbs *more* $\sim$ *most* as a very complex one in its distribution. Moreover, the fact that no suffix behaves like an overpowering default choice in MGL's results suggests that the niche distribution of the German plural suffixes is also complex. Further tests will help determine to what extent this implicational model may have an advantage over purely symbolic models by being able to capture subtle distinctions between niches through its distributed representations of word forms.

Acknowledgements

This research was originally inspired by discussions of the Corkery et al. (2019) and McCurdy et al. (2020) papers at Colin Wilson's 2021 morphology seminar and by a presentation by Coleman Haley on those two papers at Paul Smolensky's Neurosymbolic Computation Lab group last spring, both at Johns Hopkins. Thanks to three anonymous reviewers and to all the members of the above two groups for helpful comments and suggestions. Thanks also to audiences at the Fifth American International Morphology Meeting where an earlier version of this research was presented. All errors are my own.

References

Farrell Ackerman, James Blevins, and Robert Malouf. 2009a. *Analogy in Grammar: Form and Acquisition*, chapter Implicative Relations in Word-Based Morphological Systems. Oxford University Press.

Farrell Ackerman, James Blevins, and Robert Malouf. 2009b. *The Oxford Handbook of Morphological Theory*, chapter Word and Paradigm Morphology. Oxford University Press.

Farrell Ackerman and Robert Malouf. 2016. *Cambridge Handbook of Morphology*, chapter Implicative Relations in Word-Based Morphological Systems. Cambridge University Press, Cambridge.

Adam Albright and Bruce Hayes. 2003. Rules vs. Analogy in English Past Tenses: A Computational/Experimental Study. *Cognition*, 90:119–161.

Mark Aronoff. 2021. Three ways of looking at morphological rivalry. Keynote talk at the 5th American International Morphology Meeting.

Sacha Beniamine and Matías Guzmán Naranjo. 2021. Multiple alignments of inflectional paradigms. In *Proceedings of the Society for Computation in Linguistics*, volume 4.

Mathieu Bernard. phonemizer.

Olivier Bonami and Sacha Beniamine. 2016. Joint predictiveness in inflectional paradigms. *Word Structure*, 9(2):156–182.

Basilio Calderone, Nabil Hathout, and Olivier Bonami. 2021. Not quite there yet: Combining analogical patterns and encoder-decoder networks for cognitively plausible inflection. Technical report, arXiv.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? Encoder-decoder neural networks as cognitive models of English past tense inflection. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3868–3877.

Gambolputty. dewiki-wordrank.

Institut für Deutsche Sprache. 2014. Korpusbasierte Wortformenliste DeReWo, DeReKo-2014-II-MainArchive-STT.100000. www.ids-mannheim.de/derewo.

Pennti Kanerva. 1988. *Sparse Distributed Memory*. MIT Press, Cambridge, MA.

Pentti Kanerva. 2009. Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. *Cognitive Computation*, 1(2):139–159.

Pentti Kanerva. 2017. Stanford Seminar - Computing with High-Dimensional Vectors. Youtube talk at https://www.youtube.com/watch?v=zUCoxhExe0o.

Christo Kirov and Ryan Cotterell. 2018. Recurrent Neural Networks in Linguistic Theory: Revisiting Pinker and Prince (1988) and the Past Tense Debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Gary F. Marcus, Ursula Brinkmann, Harald Clahse, Richard Wiese, and Steven Pinker. 1995. German inflection: The exception that proves the rule. *Cognitive psychology*, 29(3):189–256.

Kate McCurdy, Sharon Goldwater, and Adam Lopez. 2020. Inflecting When There's No Majority: Limitations of Encoder-Decoder Neural Networks as Cognitive Models for German Plurals. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1745–1756.

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

Sarah Payne, Caleb Belth, Jordan Kodner, and Charles Yang. 2021. The Recursive Search for Morphological Productivity. Poster presented at the 5th American International Morphology Meeting.

Dominic Schmitz, Ingo Plag, and Dinah Baer-Henney. 2021. Reconsidering pseudowords in morphological research. Slides for talk at the 5th American International Morphology Meeting.

Paul Smolensky. 1990. Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems. *Artifical Intelligence*, 46:159–216.

Jochen Trommer. 2021. The subsegmental structure of German plural allomorphy. *Natural Language and Linguistic Theory*, 39:601–656.

Charles Yang. 2016. *The price of Linguistic Productivity: How Children Learn to Break the Rules of Language*. MIT Press.

Parsing Early Modern English for Linguistic Search

Seth Kulick and **Neville Ryant**
Linguistic Data Consortium
University of Pennsylvania
{skulick,nryant}@ldc.upenn.edu

Beatrice Santorini
Linguistics Dept.
University of Pennsylvania
beatrice@sas.upenn.edu

Abstract

This work addresses the question of whether the output of a state-of-the-art parser is accurate enough to support research in theoretical linguistics. In order to build reliable models of syntactic change, we aim to eventually parse the 1.5-billion-word Early English Books Online (EEBO) corpus. But since EEBO is not yet parsed, we begin by constructing and testing a parser on the 1.7-million-word Penn-Helsinki Parsed Corpus of Early Modern English (PPCEME). In order to obtain robust results, we define an 8-fold split on PPCEME. We then evaluate the parser with evalb and, more relevantly for us, with a task-specific metric - namely, its accuracy in parsing 6 sentence types necessary to track the rise of auxiliary *do* (as in *They did not come* vs. its historical precursor *They came not*). Retrieving the relevant sentences from the gold and test versions with CorpusSearch queries (Randall, 2010), we find that the parser's accuracy promises to be sufficient for our purposes. A remaining concern is the variability of the output, which we plan to address with three pieces of future work sketched in the conclusion.

1 Introduction

The Penn-Helsinki Parsed Corpus of Early Modern English (PPCEME) (Kroch et al., 2004) consists of over 1.7 million words of text from 1500 to 1712, manually annotated for phrase structure. It belongs to a family of treebanks of historical English (Taylor et al., 2003; Kroch et al., 2000b; Taylor et al., 2006; Kroch et al., 2016) and other languages (Wallenberg et al., 2011; Galves et al., 2017; Martineau et al., 2021; Kroch and Santorini, 2021) with a shared annotation philosophy and similar guidelines across languages, which form the basis for reproducible studies of syntactic change (Kroch et al., 2000a; Ecay, 2015; Wallenberg, 2016; Galves, 2020; Wallenberg et al., 2021).

While all of these corpora are relatively large for manually annotated corpora, there are important limits on their usefulness - notably, the fact that even relatively common phenomena still occur too rarely to support reliable statistical models of how they change over time. We therefore wish to parse and search the much larger corpora that are becoming publicly available. For instance, with its 1.5 billion words of text from 1475 to 1700, the Early English Books Online (EEBO) corpus (Text Creation Partnership, 2019) dwarfs PPCEME. However, its potential as a resource for linguistic research remains unrealized because it is not linguistically annotated and its size renders manual annotation infeasible. Our eventual goal is therefore to parse EEBO automatically.

This paper reports on a first step in that direction - namely, building a parser whose accuracy we can evaluate on the gold standard provided by PPCEME. For our purposes, the standard evaluation metric, evalb (Sekine and Collins, 2008), is not specific enough. Evaluation measures based on joint effects of parser output with other factors are also inappropriate, since retrieving the sentence types of interest to us is a direct function of the parse, without any intervening processing. It is clear that the most useful evaluation metric for our purposes involves scoring the retrieval of the diagnostic sentence types. Here, we report on negative declarative sentences, on negative imperatives, and on direct questions, each in two variants. The first variants are the ones that were dominant in 1500 (*They drank not the ale, Drink not the ale, Drank they the ale?*), and the second ones are their modern counterparts, which had become dominant by 1700 (*They did not drink the ale, Do not drink the ale, Did they drink the ale?*). We choose these sentence types because we hope that large datasets like EEBO will eventually allow us to decide between different conceptual models of the change - specif-

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 143-157.
Held on-line February 7-9, 2022

ically, competition (Kroch, 1989; Zimmermann, 2017) versus drift (Karjus, 2020).

The remainder of the paper is structured as follows. Section 2 discusses some features of PPCEME's source material and annotation that present challenges for state-of-the-art parsers, especially as compared to more widely used treebanks such as the Penn Treebank (PTB) (Marcus et al., 1993). Sections 3 and 4 describe our cross-validation split of PPCEME for evaluating the parser and our use of EEBO to create contextualized word embeddings for the parser. Section 5 presents the parser model, along with results based on evalb, which we include for general comparability beyond our task-specific evaluation metric. Section 6 illustrates the diagnostic sentence types and the queries that we use for retrieving them, which are formulated in the CorpusSearch query language (Randall, 2010). Section 7 presents the results from the task-specific evaluation, and Section 8 summarizes with an eye towards future work.

2 PPCEME Issues

PPCEME differs from PTB in several important ways, making it an excellent test case for domain adaptation of modern parsing technology. However, there has been relatively little work in the NLP community using PPCEME and its sister corpora, the Penn Parsed Corpus of Middle English, 2nd edition (PPCME2) and the Penn Parsed Corpus of Modern British English (PPCMBE).[1] Kulick et al. (2014) describe parsing PPCMBE, while Moon and Baldridge (2007) and Yang and Eisenstein (2016) focus on POS-tagging (the former on PPCME2, and the latter on PPCEME and PPCMBE).

In addition to the nonstandard orthography and the different and variable syntax of the source material, PPCEME is annotated according to guidelines arising in part from its purpose for linguistic research that require explicit consideration.

2.1 PPCEME Part-of-Speech Tags

2.1.1 Complex Tags

Although we generally attempt to avoid modifying the existing annotation, PPCEME's very large set of POS tags (N = 353) requires trimming to a computationally more tractable size.

Of the 353 tags just mentioned, 213 are complex tags intended to facilitate tracking changes in orthographic conventions over time - for instance, the development of `(ADJ gentle) (NS men)` to `(ADJ+NS gentlemen)`. Since these changes are irrelevant for present purposes, we prune such tags in accordance with the Righthand Head Rule, yielding `(NS gentlemen)`.[2] Certain rare cases, such as `(WPRO+ADV+ADV whatsoever)` or `(Q+BEP+PRO albeit)`, are exceptions to the Righthand Head Rule. In such cases, the best simple tag is sometimes the leftmost tag and sometimes another tag entirely (`(WPRO whatsoever)`, `(P albeit)`). We simply ignore this complication on the grounds that these cases are a small subset of the complex tags, which themselves are used for only about 1% of the words in the corpus. After pruning and some other minor changes discussed in Appendix A, 85 POS tags remain.

2.1.2 Distinctions among Verb Classes

PTB makes no distinction between main verbs and the auxiliary verbs *be*, *do* and *have*, but this distinction is vital for us, since it is exactly the syntax of main (but not auxiliary) verbs that changes over the course of Early Modern English. In fact, even among the verbs with auxiliary uses, we need to distinguish *do* from the other auxiliaries in order to track the rise of auxiliary *do*. For this reason, we do not follow Yang and Eisenstein (2016) in mapping the PPCEME tags for verbs to the smaller set used in PTB.

2.2 PPCEME Phrase Structure

2.2.1 Function Tags

In phrase-structure treebanks, function tags can be appended to syntactic category labels in order to provide information about a constituent's grammatical or semantic role. The PTB uses 20 function tags in this way, while exploiting structural differences to distinguish other constituent roles. By contrast, PPCEME relies on function tags uniformly, largely because it has neither base NPs or VPs. As a result, PPCEME's set of function tags is larger than PTB's. Omitting a few rare types, we use 31 in the work reported below.[3] The following tree illustrates PPCEME's use of function tags to encode central grammatical roles. The subject and indirect object are sisters, but distinguished by the

[1] These two corpora and PPCEME are collected in Kroch (2020).

[2] Yang and Eisenstein (2016) simplify the complex tags for the same reason as we do, but keep the leftmost tag, which for English is incorrect in the general case.

[3] See Appendix B for the details, along with some information on function tag frequency.

function tags SBJ and OB2, respectively. MAT and SUB on the two IPs identify the higher one as a matrix clause and the lower one as a subordinate clause. Finally, THT indicates that the CP is a *that* complement clause (rather than, say, a relative or adverbial clause).

```
(IP-MAT (CONJ and)
        (NP-SBJ (D the) (N schereffe))
        (VBD shewed)
        (NP-OB2 (PRO$ my) (N servant))
        (CP-THT (C that)
                (IP-SUB ...)))
```

There has been some work on recovering function tags in PTB (Blaheta and Charniak, 2000; Blaheta, 2003; Gabbard et al., 2006; Merlo and Musillo, 2005), but overall they have received only limited attention. We are not aware of any work to recover the function tags in the historical corpora. Given the centrality of certain function tags (notably, SBJ) for retrieving the sentence types of interest to us, we are constrained to include them in the parsing model.

2.2.2 Empty Categories

PPCEME indicates discontinuous dependencies by means of empty categories that are coindexed with a displaced constituent. Following common NLP practice, we remove both the empty categories and the co-indexing from the parser training material, and thus from the parser output. This simplifies the parsing model, and for present purposes, the absence of empty categories is irrelevant. However, if we wish to include linguistic queries in future work that make reference to empty categories, as is necessary in the general case, the parsing model will need to be augmented appropriately.

3 Cross-validation Splits

Parsing work relies on train/dev/test splits of the source material used for training and evaluation. Recently, concerns have been raised over the validity of inferences drawn from static train/dev/test splits; for instance, see Gorman and Bedrick (2019), who evaluate the consistency of rankings of POS taggers across 20 random splits of the WSJ section of PTB. For us, this issue is particularly pressing because PPCEME contains relatively few individual source texts, thus increasing the chance that a single particularly difficult or non-representative source text will greatly skew performance on the dev/test partitions. Even more seriously, certain constructions might be completely absent from one particular split. This is of particular concern to us because direct questions, though common in ordinary conversation, are rare or completely absent in many written genres. We return to this point in Section 7.2.2.

We therefore define an 8-fold cross-validation split, with each component split roughly matching the 90%-5%-5% distribution in the standard single PTB split. Within each partition (train, dev, test) of a split, we attempted to equally represent (in terms of equal word counts) each of PPCEME's three time periods, as indicated by "e1", "e2", and "e3" in the filenames. Given our eventual goal of parsing all of EEBO, which encompasses all of these time periods, this step is necessary in order to adequately predict performance on that corpus.[4] Finally, in cases where PPCEME distributes a single source text over several annotated files, we were careful to assign all such files to the same partition. As PPCEME contains 448 annotated files, but only 232 distinct source texts, this greatly constrained how we could define the partitions. Nevertheless, we succeeded in including 209 (90%) of the 232 source texts in either a dev or test partition of one of the 8 splits. For more details on the split definitions, see Appendix C.

4 ELMo Embeddings Trained on EEBO

In recent years, contextualized word embeddings such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) have driven significant improvements on downstream NLP tasks, including POS tagging and parsing. Due to the significant overhead involved in training these representations, researchers often use pretrained models distributed by large companies, sometimes fine-tuned to the domain of interest. Although this often produces perfectly satisfactory results, in cases of significant mismatch between a test domain and standard training domains - usually sources such as text scraped from Wikipedia, BooksCorpus (Zhu et al., 2015), and news text from Common Crawl (Nagel, 2016) - pretraining on the novel domain yields significant improvements (Lee et al., 2019; Beltagy et al., 2019; Jin et al., 2019).

Because of the orthographic and syntactic differences between Early Modern English and contemporary English mentioned in Section 2, our current

[4] By contrast, Yang and Eisenstein (2016), split PPCEME into thirds by time period (rather than across time periods) for the different purpose of studying domain adaptation.

work involves exactly such a mismatch, and so we pretrained ELMo embeddings on EEBO.[5]

We used the same model configuration as Peters et al. (2018) for 11 epochs[6] using all of EEBO. We then integrated the resulting embeddings, which have 1,024 dimensions, into the parser model, as discussed in Section 5. Here, we describe some main aspects of creating the embeddings, which we will make public. See Appendix D for further details.

4.1 Text Extraction, Normalization, and Tokenization

EEBO's XML files contain a great deal of metadata and markup in addition to the text. For each file, we extracted the core source information (title, author, date) and kept the text within `<P>` tags, which gives at least a rough sense of the document divisions. Following Ecay (2015, pp. 105-6), we excluded some metadata and other material embedded in the text. We also adopted his handling of `GAP` tags for OCR errors, which consists of mapping these tags to word-internal bullet characters - e.g., `Eccl•siasticall`.

After normalizing the extracted text with Unicode NFC form in order to eliminate spurious surface differences between tokens, we tokenized the EEBO text in accordance with PPCEME's tokenization guidelines as best we could:

1. Possessive morphemes are not separated from their host (e.g., `Queen's`) (unlike in PTB).

2. Punctuation is separated except in the case of abbreviations (e.g., `Mr.`), token-internal hyphens (e.g., `Fitz-Morris`), or certain special cases (e.g., `&c`).

3. Roman numerals can include leading, internal, or trailing periods (e.g., `.xiiii.C.`).

PPCEME tokenization is straightforward in principle, but the non-standardized nature of the historical material raises various difficulties. For instance, it is easy to tell that the elided article `th'` should be split off (e.g., `th'exchaung` is tokenized as `th' exchaung`). But when the apostrophe is missing, the status of `th` is unclear (e.g., `thafternoone` is tokenized as `th afternoone`, but `thynkyth` remains a single token). Another example of pervasive ambiguity is `its` and `it's`; in PPCEME, these forms were tokenized manually as one token or two, depending on whether the spelling represents the possessive form of the pronoun *it* or the contracted form of *it is*. Since EEBO's size rules out manual processing, we resolved such ambiguities by defaulting to the more common case. In the above examples, this resulted in splitting the variants with apostrophes and not splitting the ones without.[7]

5 Model and Evaluation

5.1 Parser Architecture

We use the parser model of Kitaev et al. (2019), which represents a constituency tree T as a set of labeled spans (i, j, l), where i and j are a span's beginning and ending positions and l is its label. Each tree is assigned a score $s(T)$, which is decomposed as a sum of per-span scores:

$$s(T) = \sum_{(i,j,l)\in T} s(i,j,l) \tag{1}$$

The per-span scores $s(i, j, l)$ themselves are assigned using a neural network that takes a sequence of per-word embeddings as input, processes these embeddings using a transformer-based encoder (Vaswani et al., 2017), and produces a span score from an MLP classifier (Stern et al., 2017). The highest-scoring valid tree is then found using a variant of the CKY algorithm. POS tags are recovered using a separate classifier operating on top of the encoder output, which is jointly optimized with the span classifier. For more details, see Kitaev and Klein (2018). As already mentioned in Section 4, we use ELMo embeddings pre-trained on EEBO.

Our implementation is based on version 0.2.0 of the Berkeley Neural Parser[8] modified to accept ELMo.[9] We train each of the 8 models (one for each cross-validation split) for 50 epochs, using the evalb score on the dev section as our criterion for

[5] Space constraints prevent us from presenting full details here, but we find that using ELMo embeddings trained on EEBO improves evalb scores by about 2 points over the standard ELMo embeddings trained on modern English and still by about 0.5 points over BERT embeddings trained on modern English. At present, we lack the computational resources for the obvious next step of pretraining BERT embeddings on EEBO, but we are pursuing access to them.

[6] This corresponds to 2 weeks of training using 4 GTX 1080 GPUs.

[7] Future work could consider a joint tokenization-POS-tagging model.

[8] https://github.com/nikitakit/self-attentive-parser

[9] These modifications and other relevant software are available at https://github.com/skulick/emeparse.

	Parser	Part-of-Speech
dev	90.89 (1.8)	98.14 (0.7)
test	90.53 (0.7)	98.30 (0.4)

Table 1: Cross-validation Parser and Part-of-Speech Results. Each result is the mean for the relevant partition (dev or test) over the 8 splits, with the standard deviation in parentheses.

saving as the best model. For more details regarding training and hyperparameters, see Appendix E.

5.2 Function Tags

Following the approach of Gabbard et al. (2006) to function tag recovery, we do not delete function tags in preprocessing, and so nonterminals like NP-SBJ are treated as atomic units. Since the decision whether to delete is part of the preprocessing, this approach does not require modification to the parser.

5.3 Evalb Results

Table 1 gives our parsing and part-of-speech results by the standard NLP measures, combined over the 8 cross-validation splits, as scored by evalb (matching brackets for the parsing score and POS accuracy for the tagging score).[10]

The evalb parsing score falls within the general range of parsing scores for PTB, though a few points lower. As Kulick et al. (2014) point out, all of the English historical corpora lack certain brackets present in PTB (base NPs and VPs) that are relatively "easy to get", and this tends to adversely affect their parsing scores. Specifically, Kulick et al. (2014) find the f1 score for PPCMBE to be lower than for PTB by about 2 points, and we would expect that effect to carry over to PPCEME.[11]

6 Diagnostic Sentence Types and Query-based Retrieval

Having obtained a rough idea of the parser's performance from the evalb scores, we now turn to the question of greater interest to us - the evaluation of the parser in task-specific terms. Recall that we wish to identify certain sentence types that allow us to track the rise of auxiliary *do* over the course of Early Modern English. For expository reasons, we present these sentence types in reverse chronological order.[12]

6.1 Sentence Types with Auxiliary *Do*

Modern English is unusual in requiring the auxiliary verb *do* in certain sentence types, notably in negative declarative sentences, in negative imperatives, and in all direct questions (whether positive or negative).

Do-not-decl. In negative declarative sentences, the main verb appears in uninflected form. Such sentences also contain auxiliary *do* in either the present or past tense, and the negative marker *not* appears between the auxiliary and the main verb.

```
(IP-SUB (NP-SBJ (PRO they))
        (DOP do)
        (NEG not)
        (NP-MSR (Q much))
        (VB minde)
        (NP-OB1 (PRO them))
```

As the above example shows, the IP in this sentence type (and also its historical counterpart) can be either an independent matrix (MAT) clause or, as here, a subordinate (SUB) clause.

Do-not-imp. Negative imperatives are analogous, except for the IMP function tag on IP, and the imperative POS tag (DOI) on the auxiliary.

```
(IP-IMP (PP (P For)
            (NP (NPR$ God’s)
                (N sake)))
       (DOI do)
       (NEG not)
       (VB overlay)
       (NP-OB1 (PRO me))
       (PP (P with)
           (NP (ADJ superfluous)
               (N Matter)))
    (. .))
```

Do-sbj. Finally, in direct questions, auxiliary *do* precedes the subject instead of following it, as in declaratives. This inversion occurs in both positive and negative questions, and so retrieving this sentence type relies crucially on the parser correctly identifying the subject via the SBJ function tag. In the following example, note that the annotation guidelines for PPCEME require direct questions to

[10] Evalb removes tokens (punctuation) from consideration based on their POS tags, and since our model predicts POS tags, this can result in inconsistent sentence lengths for the gold and parsed trees if there are POS tag errors, resulting in "Error" sentences in the evalb output. We therefore use the modified evalb supplied with the Berkeley parser, due to Seddah et al. (2014), which does not delete any words, so that any POS tag differences have no effect on sentence length.

[11] For some discussion of function tag accuracy from an NLP perspective, see Appendix F.

[12] We are concerned only with sentences without modal verbs (*can*, *will*, etc.), aspectual auxiliaries *have* and *be*, or main verb *be*; sentences containing these elements were not affected by the change.

be annotated as CP-QUE-MAT immediately dominating IP-SUB. In this context, the IP-SUB is understood as part of the direct question rather than an ordinary subordinate clause.

```
(CP-QUE-MAT (WADVP (WADV How))
    (IP-SUB (DOP do's)
            (NP-SBJ (D this) (N Sute))
            (VB fit)
            (NP-OB1 (PRO me)))
    (NP-VOC (NPR Dauy))
    (. ?))
```

6.2 Sentence Types Without Auxiliary *Do*

We now illustrate the historical precursors of the modern sentence types just discussed. In all 3 old forms, it is the main verb (rather than auxiliary *do*) that appears in a past or present tense form, and it occupies the same position as auxiliary *do*. Thus, we have negative declarative sentences (`verb-decl-not`) like:

```
(IP-SUB (NP-SBJ (PRO I))
        (VBD sent)
        (NEG not)
        (PP (P to)
             (NP (PRO you))))
```

negative imperatives (`verb-not-imp`) like:

```
(IP-IMP (VBI let)
        (NEG not)
        (IP-INF (NP-SBJ (D that))
                (VB hurt)
                (NP-OB1 (PRO me)))
    (. .))
```

and questions (`verb-sbj`) like:

```
(CP-QUE-MAT
    (WADVP (WADV When))
    (IP-SUB (VBP comes)
            (NP-SBJ (PRO$ your)
                    (N Taylor))
            (ADVP-DIR (ADV hither)))
    (. ?))
```

6.3 Sample CorpusSearch Query

In order to retrieve the 6 diagnostic sentence types, we formulate queries in CorpusSearch (Randall, 2010), a query language for querying, editing, and coding tree structures. Each query is a sequence of boolean conditions on the parser output. For instance, the following query retrieves direct questions with auxiliary *do* (`do-sbj`).

```
    (CP-QUE-MAT* iDoms IP-SUB*)
AND (IP-SUB* iDoms DOD|DOP)
AND (IP-SUB* iDoms NP-SBJ*)
AND (IP-SUB* iDoms DO|VB)
AND (DOD|DOP precedes NP-SBJ*)
AND (NP-SBJ* precedes DO|VB)
```

The asterisks on the labels allow the query to match tokens with further trailing function tags (say, -SPE to indicate direct speech or -RSP for resumptive subjects). In concluding this section, we draw the reader's attention to the fact that our queries are all formulated assuming that the parser has constructed the relevant clause boundaries correctly. In Section 7.2.1, we discuss an attempt to improve parser performance by allowing structures without IP-SUB or with a recursive IP-SUB to count as matches.

7 Query-Based Results and Analysis

7.1 Results

We evaluated the parser in task-specific terms as follows. For each split, we (1) trained the parser on that split's training section and (2) parsed the split's dev and test sections. We then (3) ran 6 CorpusSearch queries, one for each of the diagnostic sentence types just presented, over the parsed sections. Cases where the queries retrieved "hits" in both the gold and the parsed tree were matches. Hits in the gold, but not the parsed tree, were classified as misses. The converse case of hits in the parsed tree, but not in the gold, were false alarms. From the results for these categories, we calculated the recall, precision, and f-measure for each split.

We calculated the mean and standard deviation of the recall, precision and f-measure over each split's dev section and over each split's test section. These results are shown in Table 2, with the associated standard deviations in parentheses. For each query, we also include the number (#) of hits in the gold version of the trees. These results are analogous to the cross-validation results using evalb in Table 1, but with the individual cross-validated query-based scores instead of the evalb metric.

7.2 Analysis

The overall f1 scores based on the queries are for the most part high enough for the overall project to remain promising. We neither expect complete parser accuracy, nor do we require it, since we can include an estimated error rate in any statistical models that we build.

However, the results exhibit a degree of variability that calls for investigation. The standard deviations are all higher than for the evalb results in Table 1, even for negative declarative sentences (the best case). This follows from the relative sparseness of the diagnostic structures in the corpus, as

	dev				test			
query	#	recall	prec	f1	#	recall	prec	f1
Negative declarative sentences								
do-not-decl	338	94.97 (3.7)	98.92 (1.7)	96.86 (1.9)	405	93.39 (4.3)	98.40 (2.3)	95.74 (1.9)
verb-not-decl	717	93.79 (4.9)	93.71 (3.3)	93.72 (3.8)	653	92.94 (4.0)	93.42 (3.4)	93.10 (2.5)
Negative imperative sentences								
do-not-imp	41	72.37 (45.3)	71.72 (44.8)	71.83 (44.7)	23	77.71 (34.1)	87.50 (35.4)	81.83 (34.0)
verb-not-imp	120	86.03 (10.4)	91.91 (7.4)	88.22 (4.2)	142	75.61 (20.0)	92.10 (6.8)	82.24 (14.9)
Questions								
do-sbj	564	89.29 (6.3)	98.32 (2.3)	93.47 (3.8)	329	84.48 (16.5)	93.75 (17.7)	86.57 (13.0)
verb-sbj	387	81.01 (13.2)	95.39 (3.8)	87.10 (8.1)	190	69.68 (19.7)	87.29 (10.9)	75.67 (14.4)
Augmented questions								
do-sbj+	564	92.23 (5.8)	98.36 (2.3)	95.10 (3.4)	329	85.92 (16.2)	93.75 (17.7)	87.44 (13.3)
verb-sbj+	387	84.16 (11.9)	94.00 (5.1)	88.35 (7.0)	190	74.39 (19.6)	83.10 (11.2)	76.26 (13.0)

Table 2: Query-based Results for the Dev and Test Sections. The first 6 sentence types are illustrated in Section 6. Augmented questions are discussed in Section 7.2.1.

compared to the much higher number of brackets evaluated by evalb. We turn now to two dimensions of this variability.

7.2.1 Recall vs. Precision and Parser Errors

In general, the recall results are lower than the precision results across all sentence types. By examining recall errors in the dev section, we have identified two of the more frequent error types.[13]

The first is an unfortunate tendency for the parser to produce nonsensical structures rather than to build parenthetical clauses. For example, for this gold question:

```
(CP-QUE-MAT
    (IP-SUB (IP-MAT-PRN (NP-SBJ (PRO I))
                        (VBP pray)
                        (NP-OB2 (PRO you)))
            (VBP speketh)
            (NP-SBJ (PRO he))
            (PP (P vnto)
                (NP (PRO vs)))))
```

the parser generates a flat structure with two subjects and two finite verbs, which is neither reasonable nor found in the training data (nor, for that matter, in the entire corpus).[14]

```
(CP-QUE-MAT
    (IP-SUB (NP-SBJ (PRO I))
            (VBP pray)
            (NP-OB2 (PRO you))
            (VBP speketh)
            (NP-SBJ (PRO he))
            (PP (P vnto)
                (NP (PRO vs)))))
```

A second problem that became apparent in connection with questions, both with and without auxiliary *do*, is that the parser sometimes violates the PPCEME's annotation guidelines by either omitting IP-SUB under CP-QUE-MAT or adding a redundant one. For example, instead of the gold

```
(CP-QUE-MAT (WNP (WPRO What)
            (IP-SUB (ADVP (ADV then))
                    (VBP think)
                    (NP-SBJ (PRO you))))
```

the parser omits the IP-SUB node:

```
(CP-QUE-MAT (WNP (WPRO What))
            (ADVP (ADV then))
            (VBP think)
            (NP-SBJ (PRO you)))
```

In this case, the parser's miss actually contains enough information for us to identify the output as an instance of `verb-sbj`. We therefore wrote 4 queries to retrieve systematic errors of this sort (missing vs. superfluous IP crossed with presence vs. absence of auxiliary *do*). Combining the results obtained in this way with the original results for questions tends to yield modest score improvements, as shown in the last two rows of Table 2 labeled `do-sbj+` and `verb-sbj+`.

7.2.2 Differences in Dev and Test Results

The results for the questions are significantly higher for the dev than for the test section. Since the dev section was used in training to determine the best model, as mentioned in Section 5.1, it might be thought that the results are naturally biased in favor of that section. But this idea is not consistent with the results in Table 1, where the evalb scores for the dev and test sections are quite similar.

A closer look at Table 2 suggests that the discrepancy between the dev and the test scores is an artifact of how the sentences types are distributed

[13] Future work could benefit from adapting the parser error analysis technique in Kummerfeld et al. (2012).

[14] It may be worth noting that parentheticals are encoded by a PRN function tag in PPCEME, whereas PTB encodes them by a separate PRN node. It may be worth investigating whether the PTB convention would improve accuracy in the cases at hand.

split	#	recall	prec	f1
dev				
0	43	67.44	90.62	77.33
1	36	94.44	94.44	94.44
2	51	74.51	97.44	84.44
3	49	67.35	91.67	77.65
4	22	95.45	91.30	93.33
5	3	66.67	100.00	80.00
6	84	89.29	98.68	93.75
7	99	92.93	98.92	95.83
test				
0	29	79.31	79.31	79.31
1	36	75.00	93.10	83.08
2	38	63.16	75.00	68.57
3	15	53.33	72.73	61.54
4	3	33.33	100.00	50.00
5	17	94.12	84.21	88.89
6	17	70.59	100.00	82.76
7	35	88.57	93.94	91.18

Table 3: Breakdown of the `verb-sbj` Scores for the Dev and Test Sections. The # of occurrences adds up to 387 for the dev section and 190 for the test section.

in the dev/test sections of each split. In contrast to the questions, the dev and test scores for negative declaratives are roughly equal. The numbers of tokens in each split are well-balanced across the two sections, and the standard deviations are low by comparison to the other sentence types. For the other queries, though, the number of gold structures is either very low (negative imperatives) or badly distributed across the dev/test sections (questions). In both cases, a better result correlates with a lower standard deviation rather than with a consistently better result on either the dev or the test section. For `do-not-imp`, the test section has a higher score (81.83) than the dev section (71.83), and its standard deviation, though still high (34.0), is lower than that for the dev (44.7). By contrast, for `verb-not-imp`, it is the dev section that has a higher score (88.22 vs. 82.24) with a lower standard deviation (4.2 vs. 14.9). This pattern is repeated for the questions (both `do-subj` and `verb-sbj`), where the dev sections have higher scores and lower standard deviations than do the test sections.

A more detailed look at the `verb-sbj` scores in the dev section sheds even further light on the matter. Table 3 breaks down the scores for the dev and test sections by split. The dev results benefit from the scores for splits 6 and 7, which are both relatively numerous and high-scoring. In particular, in split 7, the dev section contains excerpts from the New Testament (authnew-e2) and a transcript of a trial (oates-e3). In both of these source texts, questions occur at a higher rate than they do in other sources, and they tend to be simple questions without parentheticals (unlike those sort discussed in Section 7.2.1). In other words, split 7 contains many easy questions.

8 Conclusion

We have presented the first results on parsing PPCEME, defined an 8-fold cross-validation split, and evaluated the parser using a query-based measure connected to an overarching project in theoretical linguistics. The precision scores are generally very good, but we identified some of the problematic structures for recall and noted that even with the use of cross-validation for evaluation, the results are highly variable. In future work, we plan to use BERT embeddings and experiment with different parser models to improve parser accuracy, even though, as noted in Section 7.2, we are able to accept limits on parser accuracy for our purposes, as long as the parser's erorrs are unbiased. It is possible that parsers based on well-defined grammatical structures, such as Flickinger (2011) or (Kasai et al., 2018) will eliminate the nonsensical structures discussed in Section 7.2.1. Another alternative, in a different direction, is to use sentence embeddings derived from word embeddings, as in Arora et al. (2017), to identify the desired sentences directly, without using a parser at all.

At the same time, we recognize that the high variability revealed by our cross-validation procedure calls for evaluation on an extended set of diagnostic sentences - a task that we plan to tackle in three ways:

(1) We will extend query-based precision testing to Santorini (2021), a corpus of roughly 325,000 words of Early Modern English consisting entirely of diagnostic sentence types.

(2) We will further extend query-based testing to a representative sample of EEBO. Though we have no gold trees for EEBO, we can evaluate precision by manually checking the query hits found in the sample. This will also allow us to compare parser performance across EEBO and PPCEME. While we would expect roughly similar scores, it would not be surprising to find a decline in accuracy due in part to the tokenization approximations and OCR errors in EEBO mentioned in Section 4.

(3) In the latter case, we find ourselves in a position to give a quite rigorous quantitative estimate of the size of such a decline. As it turns out, about 40% of PPCEME overlaps roughly in underlying source

text with EEBO, and we have carried out a word alignment between the parallel texts. Thus, after training the parser on the non-overlapping 60% of PPCEME and running our queries on the parser output for both parallel texts, comparing the query-based results should give us the desired estimate for any performance dropoff to be expected when parsing the rest of EEBO.

Acknowledgments

This work was supported by NSF grant BCS 13-046668, ”Annotating and extracting detailed syntactic information from a 1.1-billion-word corpus.” We thank the three anonymous reviewers for their comments. We would also like to acknowledge here the tremendous debt, intellectual and in other ways, that this work owes to the late Prof. Anthony Kroch.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*.

Don Blaheta. 2003. *Function Tagging*. Ph.D. thesis, Brown University.

Don Blaheta and Eugene Charniak. 2000. Assigning function tags to parsed text. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

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.

Aaron Ecay. 2015. *A multi-step analysis of the evolution of English do-support*. Ph.D. thesis, University of Pennsylvania.

Dan Flickinger. 2011. Accuracy vs. robustness in grammar engineering. In E. M. Bender and J. E. Arnold, editors, *Language from a cognitive perspective: Grammar, usage, and processing*, CSLI Publications, pages 31–50. Stanford.

Ryan Gabbard, Seth Kulick, and Mitchell Marcus. 2006. Fully parsing the Penn Treebank. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 184–191, New York City, USA. Association for Computational Linguistics.

Charlotte Galves. 2020. Relaxed V-Second in Classical Portuguese. In Rebecca Woods and Sam Wolfe, editors, *Rethinking Verb-Second*, pages 368–395. Oxford University Press.

Charlotte Galves, Aroldo Leal de Andrade, and Pablo Faria. 2017. Tycho Brahe Parsed Corpus of Historical Portuguese. http://www.tycho.iel.unicamp.br/~tycho/corpus/texts/psd.zip.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791.

Qiao Jin, Bhuwan Dhingra, William W Cohen, and Xinghua Lu. 2019. Probing biomedical embeddings from language models. *arXiv preprint arXiv:1904.02181*.

Andres Karjus. 2020. *Competition, selection and communicative need in language change: An investigation using corpora, computational modelling and experimentation*. Ph.D. thesis, University of Edinburgh.

Jungo Kasai, Robert Frank, Pauli Xu, William Merrill, and Owen Rambow. 2018. End-to-end graph-based tag parsing with neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1181–1194.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Anthony Kroch. 1989. Reflexes of grammar in patterns of language change. *Language Variation and Change*, 1(3):199–244.

Anthony Kroch. 2020. Penn Parsed Corpora of Historical English. LDC2020T16 Web Download. Philadelphia: Linguistic Data Consortium. Contains Penn-Helsinki Parsed Corpus of Middle English, second edition, Penn-Helsinki Parsed Corpus of Early Modern English, and Penn Parsed Corpus of Modern British English.

Anthony Kroch and Beatrice Santorini. 2021. Penn-BFM Parsed Corpus of Historical French, version 1.0. https://github.com/beatrice57/mcvf-plus-ppchf.

Anthony Kroch, Beatrice Santorini, and Lauren Delfs. 2004. Penn-Helsinki Parsed Corpus of Early Modern English (PPCEME). CD-ROM, first edition, release 3. http://www.ling.upenn.edu/ppche/ppche-release-2016/PPCEME-RELEASE-3.

Anthony Kroch, Beatrice Santorini, and Ariel Diertani. 2016. Penn Parsed Corpus of Modern British English (ppcmbe2). CD-ROM, second edition, release 1. http://www.ling.upenn.edu/ppche/ppche-release-2016/PPCMBE2-RELEASE-1.

Anthony Kroch, Ann Taylor, and Donald Ringe. 2000a. The Middle English verb-second constraint: A case study in language contact and language change. In Susan Herring, Lene Schoessler, and Peter van Reenen, editors, *Textual parameters in older language*, pages 353–391. Benjamins.

Anthony Kroch, Ann Taylor, and Beatrice Santorini. 2000b. Penn-Helsinki Parsed Corpus of Middle English (PPCME2). CD-ROM, second edition, release 4. http://www.ling.upenn.edu/ppche/ppche-release-2016/PPCME2-RELEASE-4.

Seth Kulick, Anthony Kroch, and Beatrice Santorini. 2014. The Penn Parsed Corpus of Modern British English: First parsing results and analysis. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 662–667, Baltimore, Maryland. Association for Computational Linguistics.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the Wall Street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059, Jeju Island, Korea. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*.

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

France Martineau, Paul Hirschbühler, Anthony Kroch, and Yves Charles Morin. 2021. MCVF Corpus, parsed, version 2.0. https://github.com/beatrice57/mcvf-plus-ppchf.

Paola Merlo and Gabriele Musillo. 2005. Accurate function parsing. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 620–627, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Taesun Moon and Jason Baldridge. 2007. Part-of-speech tagging for Middle English through alignment and projection of parallel diachronic texts. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 390–399, Prague, Czech Republic. Association for Computational Linguistics.

Sebastian Nagel. 2016. Cc-news. https://commoncrawl.org/2016/10/news-dataset-available/.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Beth Randall. 2010. CorpusSearch 2: a tool for linguistic research. Download site: http://corpussearch.sourceforge.net/CS.html. User guide: https://www.ling.upenn.edu/~beatrice/corpus-ling/CS-users-guide/index.html.

Beatrice Santorini. 2021. Parsed Ellegård 1953 dataset. https://github.com/beatrice57/ellegard-examples-parsed.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 103–109, Dublin, Ireland. Dublin City University.

Satoshi Sekine and Michael Collins. 2008. evalb. http://nlp.cs.nyu.edu/evalb/.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistic (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Ann Taylor, Arja Nurmi, Anthony Warner, Susan Pintzuk, and Terttu Nevalainen. 2006. Parsed Corpus of Early English Correspondence. Distributed through the Oxford Text Archive.

Ann Taylor, Anthony Warner, Susan Pintzuk, and Frans Beths. 2003. York-Toronto-Helsinki Parsed Corpus of Old English Prose. Distributed by the Oxford Text archive.

Text Creation Partnership. 2019. Early English Books Online. https://textcreationpartnership.org/tcp-texts/eebo-tcp-early-english-books-online/. Version 2019-04-25.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Joel C. Wallenberg. 2016. Extraposition is disappearing. *Language*, 92(4):e237–e256.

Joel C. Wallenberg, Rachael Bailes, Christine Cuskley, and Anton Karl Ingason. 2021. Smooth signals and syntactic change. *Languages*, 6(2):60.

Joel C. Wallenberg, Anton Karl Ingason, Einar Freyr Sigurdhsson, and Eiríkur Rögnvaldsson. 2011. Icelandic Parsed Historical Corpus (IcePaHC), v0.9. http://www.linguist.is/icelandic_treebank.

Yi Yang and Jacob Eisenstein. 2016. Part-of-speech tagging for historical English. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1318–1328, San Diego, California. Association for Computational Linguistics.

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 *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Richard Zimmermann. 2017. *Formal and quantitative approaches to the study of syntactic change: Three case studies from the history of English*. Ph.D. thesis, University of Geneva.

A PPCEME Part-of-Speech Modifications

In addition to the changes described in the main text, we changed the tag `MD0` to `MD`. `MD0` is an untensed modal, as in `he will can` or `to can do something`. There are only 4 cases, as this is an option that had mostly died out by the time of Early Modern English.

There are also cases where words that are ordinarily spelled as a single orthographic token are sometimes split into several tokens. PPCEME represents the former case with a single POS tag and the latter as a constituent whose non-terminal is the POS tag, with the words given numbered segmented POS tags - for example, `(ADJ alone)` vs. `(ADJ (ADJ21 a) (ADJ22 lone))`. We modified all such tags by removing the numbers, and appending `_NT` to the nonterminals, in order to more clearly distinguish between POS tags and nonterminals. In this example, the resulting structure would be `(ADJ_NT (ADJ a) (ADJ lone))`.

B PPCEME Issues

B.1 Metadata

In addition to the changes described in Section 2.2.1, we removed the metadata under `CODE`, `META`, and `REF` nodes. In cases where `CODE` dominated a leaf, removing the leaf resulted in an ill-formed tree. The 267 trees in question were removed, as were 576 trees rooted in `META` (usually stage directions for a play) and 9 trees containing `BREAK`.

In addition, before carrying out the above modifications, we changed all instances of `(CODE <paren>)` and `CODE <$$paren>)` to `(OPAREN -LRB-)` and `(CPAREN -RRB-)`, respectively. We did this in order to retain the parentheses that otherwise, being daughters of `CODE`, would have been deleted.

Our counts of number of words and sentences differ slightly from Yang and Eisenstein (2016). We aim to resolve these discrepancies, which are probably related to small differences of preparation of the type just discussed.

B.2 Function Tags

We exclude certain tags that occur very rarely in PPCEME (ADT, CLF, COM, ELAB, EXL, RFL, TAG, TMC, TPC, XXX, YYY). Table 4 shows the frequency for each of the remaining 31 tags in the entire corpus, for nonterminals with a non-empty yield. For convenience, the tags are organized into six groups. The syntactic and semantic groups are roughly similar to those groups for the PTB, as presented in Gabbard et al. (2006). The other groups include tags that differ significantly from those in the PTB, as noted in Section 2.2.1. For the full set of PPCEME function tags, see `https://www.ling.upenn.edu/hist-corpora/annotation/labels.htm`.

Tag	Description	Frequency
	Syntactic	**37.23**
SBJ	subject	21.00
OB1	direct object	11.96
OB2	indirect object	1.20
SPR	secondary predicate	0.28
MSR	measure	1.17
POS	possessive	0.86
VOC	vocative	0.77
	Semantic	**7.93**
DIR	directional	0.50
LOC	locative	0.84
TMP	temporal	3.09
ADV	adverbial	3.50
	CP only	**8.83**
CAR	clause-adjoined	0.55
REL	relative clause	3.36
THT	THAT clause	2.52
CMP	comparative	0.53
QUE	question	1.35
FRL	free relative	0.33
EOP	empty operator	0.19
	IP only	**9.67**
INF	infinitive	4.59
PPL	participial	2.18
IMP	imperative	1.12
SMC	small clause	0.90
PRP	purpose	0.46
ABS	absolute	0.42
	CP or IP	**33.10**
SUB	subordinate	14.52
MAT	matrix	12.66
SPE	direct speech	5.64
DEG	degree	0.28
	Miscellaneous	**3.23**
PRN	parenthetical	2.60
RSP	resumptive	0.33
LFD	left-dislocated	0.30

Table 4: Function Tags in PPCEME with Their Frequencies. The tags are organized into 6 groups, with combined frequency by group in boldface.

C Train/Dev/Test Split

Table 5 summarizes the composition of the train/dev/test sections across the cross-validation 8 splits; specifically, the total number of documents, the total number of tokens, and the percentage of total tokens in each section. Since the

section	# files		# tokens		% of split	
train	205.88	(13.34)	1743211.25	(10441.53)	89.65	(0.54)
dev	12.50	(7.15)	101000.12	(4081.82)	5.19	(0.21)
test	13.62	(7.91)	100268.62	(7832.66)	5.16	(0.40)
OVERALL	232	(0.00)	1944480	(0.00)	100	(0.00)

Table 5: Mean number of files and tokens for train/dev/test sections across the 8 cross-validation splits (standard deviations are presented in parentheses). The percentage of tokens in each section is also presented (in the **% of split** column).

partitioning process is performed at the level of PPCEME source files, and these files differ substantially in size, there is some variation in these numbers across the splits. For this reason, we report standard deviations as well as means. The final row ("OVERALL") gives numbers for a complete split (i.e., the train/dev/test sections combined); as these are constant across each split, the entries in this row have a standard deviation of zero. As can be seen, overall the splits attain the target 90-5-5 breakdown; e.g., the train section on average comprises 89.65% of the total tokens with a standard deviation of 0.54%.

As mentioned in the main text, the corpus consists of text from three main time periods (e1, e2, e3),[15] and we aimed to balance the time periods equally within each split, to the extent possible given that we treated the files as atomic units. Table 6 shows the breakdown by period. Similar to Table 5, mean/standard deviation for total number of documents/tokens are presented for each time period in each section. Additionally, for each time period, the table reports the mean percentage of each split (in tokens) from each time period. The marginals provide numbers combining across time periods (the "ALL PERIODS" row) and sections (the "ENTIRE SPLIT" column). For example, the training section contains on average 1,743,211.25 tokens, with on average 32.85% coming from time period e1, 36.61% from e2, and 30.53% from e3.

D ELMo Embeddings Trained on EEBO

In addition to the normalizations discussed in Section 4, we follow (Ecay, 2015) in removing information under NOTE, SPEAKER, and GAP, as well as L ("line of verse") which was not appropriate for our searches. In future work, we will likely revise this to keep the text but with some meta-tags to indicate its origin.

The extracted text underwent Unicode normalization to NFC form in order to eliminate spurious surface differences between tokens. The resulting text contained 642 unique characters, 381 of which occurred fewer than 200 times. Manual inspection of these uncommon characters revealed that while some of them made sense in context (e.g., within sections of Greek or Latin text), many seemed to be spurious characters due to OCR errors (e.g., WHITE RECTANGLE 0X25AD). Consequently, we elected to filter out all sentences containing characters occurring fewer than 200 times. This eliminated 4139 lines, with 9,341,966 remaining for training (consisting of 1,168,749,620 tokens).

The ELMo embeddings were trained using TensorFlow maintained and distributed by AllenNLP at https://github.com/allenai/bilm-tf using the default model configuration.

E Model and Evaluation

Table 7 shows the hyperparameter settings used in the Berkeley Neural Parser. These are all the default settings for these parameters. We added a parameter max_epochs, used to set the maximum number of epochs. For the cross-validation training reported, we set max_epochs=50.

F Function Tag Evaluation

Function tags are typically removed by evalb before it compares bracket labels, and we have not modified this. To evaluate function tag recovery, we follow the approach of Gabbard et al. (2006). who in turn follow Blaheta (2003). Under this approach, function tags are evaluated only for nonterminals that evalb counts as matches. For example, an NP-SBJ in the parsed tree corresponding to an NP-SBJ in the gold tree counts as a match for SBJ. But an NP-OB1 in the parsed tree corresponding to an NP-SBJ node in the gold tree (which is possible since

[15] For details regarding the PPCEME time periods (e1, e2 and e3) see https://www.ling.upenn.edu/hist-corpora/PPCEME-RELEASE-3/description.html

	train section			dev section			test section			ENTIRE SPLIT		
period	# files	# tokens	% train	# files	# tokens	% dev	# files	# tokens	% test	# files	# tokens	% split
e1	72.88	572672.62	32.85	4.25	33178.50	32.98	4.88	31369.88	31.50	82	637221	32.77
	(6.51)	(11974.31)	(0.79)	(3.01)	(7078.50)	(7.36)	(4.55)	(8193.65)	(8.67)	(0.00)	(0.00)	(0.00)
e2	66.00	638269.88	36.61	4.00	34844.62	34.40	4.00	35186.50	34.89	74	708301	36.43
	(4.38)	(13490.18)	(0.60)	(2.51)	(6382.81)	(5.41)	(2.14)	(7767.44)	(6.18)	(0.00)	(0.00)	(0.00)
e3	67.00	532268.75	30.53	4.25	32977.00	32.63	4.75	33712.25	33.60	76	598958	30.80
	(5.18)	(7066.41)	(0.35)	(3.96)	(5211.71)	(4.65)	(3.45)	(5592.81)	(4.70)	(0.00)	(0.00)	(0.00)
ALL	205.88	1743211.25	100	12.50	101000.12	100	13.62	100268.62	100	232	1944480	100
PERIODS	(13.34)	(10441.53)	(0.00)	(7.15)	(4081.82)	(0.00)	(7.91)	(7832.66)	(0.00)	(0.00)	(0.00)	(0.00)

Table 6: Mean number of files and tokens for train/dev/test sections within each of three time periods (e1, e2, and e3) across the 8 cross-validation splits. The **% train/dev/test** columns indicate the % of total train/dev/test tokens for each time period. Standard deviations are presented in parentheses.

hyperparameter	value
attention_dropout	0.2
batch_size	32
char_lstm_input_dropout	0.2
checks_per_epoch	4
clip_grad_norm	0.0
d_char_emb	64
d_ff	2048
d_kv	64
d_label_hidden	256
d_model	1024
d_tag_hidden	256
elmo_dropout	0.5
encoder_max_len	512
force_root_constituent	'auto'
learning_rate	5e-05
learning_rate_warmup_steps	160
max_consecutive_decays	3
max_len_dev	0
max_len_train	0
morpho_emb_dropout	0.2
num_heads	8
num_layers	8
predict_tags	True
relu_dropout	0.1
residual_dropout	0.2
step_decay_factor	0.5
step_decay_patience	5
tag_loss_scale	5.0
max_epochs	50

Table 7: Hyperparameters Used with the Berkeley Neural Parser.

	Function Tags
dev	94.90 ± 1.54
test	95.55 ± 0.87

Table 8: Cross-validation Function Tag Results.

the function tags do not count for evalb) counts as a recall error for SBJ and as a precision error for OB1.

Table 8 shows dev and test section scores for the function tags using this scoring method, analogously to Table 1. To explore these numbers in greater depth, Table 9 breaks down the function tag results for the first cross-validation split, organized as in Table 4. By far the most significant cause for a decreased score is the SPE tag indicating direct speech. Though one of the most common tags, with a frequency of 8.21%, it attains an f1 score of only 50.75. The discrepancy reflects the absence in PPCEME of consistent clues for direct speech (such as quotation marks) that are available, say, in modern newswire text subject to strict style guidelines. Fortunately, however, SPE is not highly relevant for the purposes at hand.

Tag	Description	Frequency	F1
Syntactic		37.03	96.57
SBJ	subject	21.57	98.06
OB1	direct object	11.63	95.33
MSR	measure	1.18	92.42
OB2	indirect object	1.06	92.10
POS	possessive	0.75	96.31
VOC	vocative	0.52	92.82
SPR	secondary predicate	0.32	76.07
Semantic		7.81	95.57
ADV	adverbial	4.38	97.03
TMP	temporal	2.43	93.44
DIR	directional	0.52	95.78
LOC	locative	0.48	93.19
CP only		8.18	91.86
REL	relative clause	3.04	92.03
THT	THAT clause	1.80	95.07
QUE	question	1.46	95.23
CAR	clause-adjoined	0.67	76.78
CMP	comparative	0.63	96.97
FRL	free relative	0.48	81.57
EOP	empty operator	0.11	88.61
IP only		9.98	95.72
INF	infinitive	4.92	98.24
PPL	participial	2.35	98.59
IMP	imperative	1.26	90.83
SMC	small clause	0.83	95.68
PRP	purpose	0.46	75.98
ABS	absolute	0.17	69.92
CP or IP		34.38	88.24
SUB	subordinate	14.94	98.72
MAT	matrix	10.92	98.07
SPE	direct speech	8.21	50.75
DEG	degree	0.31	86.31
Miscellaneous		2.62	81.45
PRN	parenthetical	1.77	87.99
RSP	resumptive	0.47	58.62
LFD	left-dislocated	0.37	73.65
Total		100.00	92.83

Table 9: Function Tag Results for the Dev Section.

Remodelling complement coercion interpretation

Frederick Gietz
Department of Linguistics
University of Toronto
Toronto, ON
frederick.gietz@utoronto.ca

Barend Beekhuizen
Department of Linguistics
University of Toronto
Toronto, ON
barend.beekhuizen@utoronto.ca

Abstract

Existing (experimental and computational) linguistic work uses participant paraphrases as a stand-in for event interpretation in complement coercion sentences (e.g. *she finished the coffee* → *she finished drinking the coffee*). We present crowdsourcing data and modelling that supports broadening this conception. In particular, our results suggest that sentences where many participants do not give a paraphrase, or where many different paraphrases are given are informative about to how complement coercion is interpreted in naturalistic contexts.

1 Interpreting word meanings in context

A central aspect of pragmatic reasoning is to construe utterance meaning which is not overly given in the sentence (Grice, 1975). This paper uses crowdsourcing and computational modeling to explore the range of possible interpretations in a particular grammatical construction in which implicit meaning is (frequently) to be inferred, namely *complement coercion* sentences. These are sentences like *they finished the coffee* or *she began a book*, where the entity-type direct object is 'coerced' into an event-type interpretation applying to that direct object (e.g., 'they finished **drinking** the coffee' or 'she began **writing** the book').

The traditional treatment is that these sentences involve a case of type-shifting, where the direct object whose extension is a physical entity is instead interpreted as an event involving that direct object (Pustejovsky and Bouillon, 1995). On this account, readers leverage the lexical-semantic information of the direct object itself to arrive at a specific event (e.g., **drink** for 'they finished the coffee'). In contrast to the type-shifting account, the pragmatic account of Piñango and Deo (2016) suggests that readers instead pragmatically retrieve a relevant scale to enrich the interpretation of aspectual verbs that have entity-type direct objects. This scale can be temporal in the case of an eventive interpretation (e.g. *I sat down and began the book*) but also spatial (e.g. *The marker begins the trail*). Crucially for our purposes, pragmatic enrichment is not a lexical process resolving a specific verb. This enrichment can take place to a greater or lesser extent, in principle even allowing for a lack of enrichment, although that option is not presented explicitly in their paper.

Complement coercion has drawn attention from different communities of scholars. Psycholinguists found that (simply put) complement coercion incurs a processing cost (McElree et al., 2001; Traxler et al., 2002), while computational linguists have shown an interest in complement coercion as a challenging case of automatically retrieving implicit aspects of sentence interpretation (Lapata and Lascarides, 2003; Roberts and Harabagiu, 2011; Zarcone et al., 2012; Chersoni et al., 2017). Interestingly, both lines of research inherit the assumption from the type-shifting account that complement coercion sentences have a verb paraphrase that represents the interpreted event, and largely design test items based on this assumption. In our paper, we focus on the computational task of modeling the interpretation of sentences containing complement coercion and the light it can shed on the two theoretical accounts, but we briefly touch on the implications for experimental work in §7.

For computational formulations of the task of complement coercion interpretation, inheriting the 'obligatory (semantic) resolution' property from the type-shifting account means that coercion interpretation is conceptualized as a case of multi-label classification in which models predict a single event label (verb) which is then evaluated against annotator consensus about the correct event label (Lapata and Lascarides, 2003; Zarcone and Padó, 2010; Zarcone et al., 2012). In §2, we demonstrate that this conception obscures many relevant and interesting cases of complement coercion where verb paraphrase is not sufficient to represent interpretation. Then, in §3 and §4, we introduce models for complement coercion interpretation designed for the simple verb prediction task. In §5 and §6, we highlight two types of cases which break from the typical examples shown in the theoretical literature – cases where participants prefer not to give any verb paraphrase ('blanks') and cases where participants are divided on which verb to use ('low-consensus'). By building improvements to our models to handle these two cases, we suggest that complement coercion is best modeled as a form of (optional, or at least gradient) pragmatic enrichment rather than as obligatory semantic completion.

2 Elicitation study

2.1 Crowdsourcing with Blank responses

Existing experimental and computational work relies on (crowdsourced) norming data to determine how complement coercion sentences are interpreted (Zarcone and Padó, 2010; McElree et al., 2001; Traxler et al., 2002; Frisson and McElree, 2008). Comparing the sentences in experimental and computational studies with cases of complement coercion from a corpus of naturally occurring text (the Corpus of Contemporary American English, or COCA: Davies, 2009), we observed that the interpretation of naturally occurring cases often differs from the hand-crafted examples used in experimental and computational work, in particular in that hand-crafted examples typically allow for a clear verb paraphrase, often a single one, whereas naturally-occurring sentences often seem to lack this property.

This exploratory observation led us to design a new elicitation experiment in which we used naturally-occurring cases of complement coercion. First, candidate complement coercion sentences, containing an aspectual verb (*begin, end, start, finish, complete*) and a likely coerced entity object were extracted from COCA using heuristics discussed in Appendix A. 300 of the $4,583$ likely instances were sampled for an elicitation experiment, in which we asked participants to fill a blank between the verb and the direct object (e.g., *She finished ____ her book*), similar to the papers cited above. In contrast to these approaches, and in line with our expectation that not all cases readily elicit verb paraphrases, participants were instructed that blanks should be left empty if no verb was felt to fit it. Appendix B presents an example of the elicitation prompt and several participants responses. [1] Using Testable (Testable), we gathered on average 19 (range: 15–20) responses per item.

In line with our initial intuition, our data displayed a large amount of 'blank' responses. In 138/300 sentences (46%), the most common response was a blank one. The remaining 162 sentences displayed substantial variation in the degree to which participants agreed with each other. Defining the *consensus* of an item as the proportion of participants who gave the dominant response, our data displays a median consensus of 55% (IQR: 40%–74%). To illustrate: an example such as (1-a) has a similar direct object as (1-b), yet received a majority of blank responses (12/19, vs. 1/15 for the latter). Similarly, example (1-c) received 4/19 *paint* responses, versus 9/15 for example (1-b), both again with similar direct objects. In contrast, constructed cases often have a high consensus compared to naturalistic examples (e.g., example (1-d) had 58% of participants in the norming study of (Frisson and McElree, 2008) respond with *paint*).

(1) a. Lordier began ___ the painting with a very light sketch of the major shapes...
b. In 1951 he began ___ a second mural, a portrayal of St. Joseph as the master craftsman...
c. You will see the final texture effect when you click OK. You have just completed ___ your textured picture.
d. The artist began ___ the portrait in his studio in the city.

2.2 Interpretive strategies vary across cases

We believe the prevalence of blank responses and low-consensus responses is not an effect of poor annotator training or different annotator conceptions of the target event, as Elazar et al. (2020) suggest, but instead an effect of the varying demands on the pragmatic resolution of the event that different examples bring about. In cases like (1-a), the implicit event is not critical to understanding the sentence; rather, the manner of the event (*with a very light sketch ...*) is more salient to interpretation. Partici-

[1] This final 300 includes 16 (5.3%) with an inanimate subject, e.g. *A pretty bow completes the picture.* We kept these sentences to compare participant responses as they fit the complement coercion pattern by definition, recognizing they potentially form a subcategory or separate aspectual verb sense.

pants may consider the specific nature of the event to be backgrounded, and for that reason elect to leave the response blank.

Similarly, in (1-c), the act of completion seems to be the primary message of the sentence rather than the specific nature of what is completed. Unlike in (1-a), blank responses do not dominate, but participants display a lower degree of consensus about which verb to fill out than for (1-b): for (1-c), 4/15 respond with *paint*, but we also find highly similar responses like *edit* (3/15), *make, design, print* and *render* (all 1/15), that all convey a sense of creation.[2] Elazar et al. (2020) argue that such low-consensus cases potentially reflect respondents' different construals of the same situation. We propose instead that the fact that 11/15 responses reflect a general sense of creation is indicative of speakers agreeing on the broad sense of the coerced event (here: 'creation'), but disagreeing when forced to come up with a specific verb to fit that broad event.

Other cases are found in the data where no verb is dominant, but where participants still give some verb responses sharing the same broad sense. For example, both sentences (2-a) and (2-b) receive majority blank responses (12/20 and 11/19 responses, respectively), but other responses include verbs about creation like *make* and *build*. Similarly, the sentences in (3) all had the most popular response *write*, but none had it as a majority (4/20, 6/19, and 7/19 responses, respectively). Less popular responses included *publish*, *make*, and *compile*.

(2) a. Lau next inserts a set of wire filaments into the chamber... He completes ___ the setup by fitting a quartz cover on the top of the reactor.
b. Complete ___ the rig by threading a double-length of wire leader through the tube and egg sinkers.

(3) a. Together with Sky Telescope's Roger W. Sinnott, Tirion has just finished ___ a new edition of his classic Sky Atlas 2000.
b. McGruder began ___ his politically charged hip-hop comic strip for his college newspaper.
c. He did finish ___ Harvard Man - a story, he says, about sex, drugs, madness, orgasm, philosophy, and college basketball fixing.

An anonymous reviewer points out that some sentences may receive blank responses due to factors besides the event interpretation. Specifically, the fill-in-the-blank style of crowdsourcing may discourage responses which are valid verb interpretations but which cause grammaticality issues or redundancy when given overtly. For example, in sentence (4-a), 15/19 participants give a blank response, compared to 3/19 who give the verb '*install*' and 1/19 with the verb '*make*.' It is likely that some participants leave this sentence blank to avoid an ungrammatical double-gerund construction. However, a grammatically similar sentence, like (4-b), receives a 90% consensus response in *eat*. Similarly, the presence of a direct object ending in *-ing* may keep participants from presenting verbs ending in *-ing*. While we are certain that these low-level factors impact consensus rates to some extent, there are many counter-examples to such explanation, among cases like (2-a) and (3), where an explanation in terms of grammaticality or redundancy avoidance cannot be given.

(4) a. FINISHING ___ THE ROLL-OFF ROOF RAILS
b. Pauline and Juliet are finishing their grapes as they watch Hilda and Walter on the tennis court .

Overall, we take inconsistent and blank responses to be information (rather than noise) about how participants actually resolve these sentences when reading. For sentences which receive many blank responses or low-consensus among responses, we suggest that participants only resolve the interpretation to a specific verb because the task formulation forces or nudges them to do so.

This leads us to suggest a novel account for the two new types of cases introduced in this paper. For both types, the presence of complement coercion does not obligatorily lead to a particular event interpretation, as implicated by the account of Pustejovsky and Bouillon (1995). Rather, they fit better the account of Piñango and Deo (2016), where speakers are said to interpret an aspectual verb as related to some pragmatically determined scale but not necessarily to resolve that interpretation to the level of a specific event. Note that it is only this property of 'obligatory resolution' on

[2] For completeness: two further responses were blank, and *click* and *prepare* were both given once.

which we compare the two accounts – this paper does not make claims about any further differing properties of the two accounts.

Overall, we take sentences where the top response is a blank and sentences with a low consensus rate as indicative that not all cases of complement coercion need to be 'resolved' to a specific event, as labeled by a specific verb. Communication can succeed even when the interpretated event is left vague or underspecified (Frisson, 2009), and models of complement coercion interpretation should capture the proposed variation in the interpretation process, as evidenced by participants' diverse types of responses. In other words, we seek to build models which make more informative predictions than a single verb paraphrase, and we argue that re-conceptualizing the task allows us to better understand the linguistic (pragmatic) properties of complement coercion in return.

3 Modeling complement coercion

3.1 Redefining the modelling task

To recapitulate: previous work defines the modeling task of complement coercion interpretation as the prediction of a single, high-consensus verb paraphrase for a given sentence in line with the theoretical conception of complement coercion as involving obligatory resolution to the level of a particular event. Our annotation data show that only a minority of all sentences display a consensus of over 50%, and for almost half the items a blank response is dominant – two effects we argued not to be due to poor annotation or improperly trained annotators, but instead to the varying pragmatic demands on the resolution of apparent cases of complement coercion. In the following sections, we evaluate complement coercion interpretation models on our new dataset.

Crucially, the gold labels derived from our dataset differ in two ways from from those as formulated by similar tasks (e.g., Zarcone et al. (2012); Chersoni et al. (2017)): (1) the correct label for items is taken to be 'blank' if the dominant response was 'blank', and (2) all the low-consensus cases are included, using the dominant response as their label. We consider these changes to see how models that follow the 'predict-the-verb' task formulation fare on these two groups of cases in §4, after which we look at two extensions that explicitly take the varying pragmatic demands on interpretation into account in §5 and §6.

3.2 Models of complement coercion

Several models have been defined to model complement coercion detection (whether a case of an aspectual verb plus direct object is coercive or not) and interpretation (which event is to be inferred to 'fill the blank'). Existing models build on the intuition that the direct object of a sentence is uniquely informative for constructing a coercion interpretation (Lapata and Lascarides, 2003; Roberts and Harabagiu, 2011; Zarcone et al., 2012). First, we use the **Example Based Learning** model, or EBL (broadened from McGregor et al., 2017's coercion identification model). EBL is our only supervised model. For a given test sentence *S*, EBL predicts the interpretation to be the most common interpretation of training sentences that have the same direct object as *S*. For example, if 6/9 of the training sentences containing the direct object *book* have the top response *write*, 2/9 *read*, and 1/9 'blank', EBL will predict the answer *write* when presented with a test sentence containing the direct object *book*. If a direct object of a test sentence does not occur in the training set, EBL predicts a blank.

A second model, the **Co-occurrence counts** model (COOC) operates on the same intuition but leverages raw unlabeled corpus data instead of labeled training data. This model is a simple application of similar models from other verb-prediction tasks (Lenci, 2011; Zarcone et al., 2012). It assumes that the verb a particular direct object occurs with in a corpus (as a direct object) will also be the most likely coercion interpretation. More specifically, the COOC model predicts the top corpus verb for a specific direct object.

Finally, we define the **Prototype vector** model (Chersoni et al., 2017) (PROTO). For a given direct object, instead of predicting the top verb by co-occurrence in a corpus, average word vectors for the top *k* verbs that the direct object co-occurs with into a prototype vector *P*, and predict the closest verb in that vector space to *P*. Here we use pre-trained word2vec (Mikolov et al., 2013) vectors from the *gensim* implementation in Python.

All three models rest on the assumption that a specific direct object (type) will predict a single recurrent interpretation. This approach has the disadvantage that it is unable to predict different interpretations for different tokens of the same direct object (such as 'make' vs 'drink' for *finish the coffee*). As a point of comparison to the models above, we furthermore used a large language model (BERT;

Devlin et al., 2019) to predict based on the context of the entire sentence rather than the direct object alone, thus allowing for different interpretation for different tokens. We adapt BERT as a model for coercion interpretation by treating the fill-in-the-blank position as a masked token to be predicted. BERT yields a distribution of relative confidences for each item in its vocabulary when used for this task. This means for each sentence, we define our **BERT** model to predict the top verb from the top *k* items in this distribution.

3.3 Experimental set-up

For this study, we split the 300 sentences in our dataset randomly into 150 training and 150 testing sentences. Within the 150 test items, we evaluate the accuracy of the models in predicting the response (either a single verb or a blank). To assess whether model performance is the same for the different cases as discussed in §2, we also report the accuracy scores for three salient groups of test items: items with a 'blank' top response ($n = 69$), cases with a non-blank top response at or above the median of 55% consensus (High Consensus, or HC, $n = 41$), and cases with a non-blank top response below the median consensus (Low Consensus, or LC, $n = 40$). In line with Roberts and Harabagiu (2011), we skipped predictions of semantically general verbs like *have* and *say*.

4 Modelling dominant verb responses

4.1 Results & Discussion

Accuracy for each model is reported in Table 1 as the **-T** ([T]op verb predicting) models. We expect models to be unable to predict **Blanks**, as they are defined to find the top-ranked verb. Interestingly, we observe that EBL performs well on this subset of the data at .870 accuracy, compared to near-zero scores for the other models. EBL's high performance, however, seems to be an artefact of data scarcity: many direct objects in the test set do not exist in the (small) training set and therefore cannot be predicted for, so EBL performs well by accident rather than by design. The non-zero scores for the other models can similarly be attributed to a few cases in which no verb among the co-occurrence data or top model predictions could be found.

	Overall	Blanks	HC	LC
EBL-T	.620	.870	.512	.300
COOC-T	.233	.058	.439	.317
COOC-B	.480	.710	.293	.275
PROTO-T	.200	.000	.488	.250
PROTO-B	.333	.623	.073	.100
BERT-T	.327	.029	.731	.425
BERT-B	.493	.435	.682	.400

Table 1: Accuracy for 4 models and variants for the entire dataset as well as the three subsets.

Turning to the **degree of consensus** next, we see, that all models perform worse on LC items than on HC items. In line with our analysis of LC items in §2, we take this difference to be indicative of the difficulty of predicting a single specific verb when LC items may have underspecified (or possibly ambiguous) meanings. Given that other datasets use deliberately high-consensus items, we believe this furthermore illustrates the challenge of modelling when using a more naturalistic sample of complement coercion sentences. For the LC sentence in example (5) each model makes a different prediction (COOC: *plant*, PROTO: *produce*, BERT: *grow*) all of which are incorrect for the dominant answer *sow* (given by 8/20 participants). The comparable closeness of some answers, however, suggests that rethinking the modeling task might be insightful, which we will do in §6.

(5) ...for a fall crop. Start ___ seeds in pots in early to midsummer, setting out six- to eight-week-old transplants in late summer or early fall in full sun and enriched soil.

Finally, focusing on a between-model comparison, we note that BERT outscores the other models on both HC and LC items. We believe this is because (1) BERT doesn't suffer from data scarcity as much as the other models by being trained on larger amounts of data, and (2) it is a token level model and can thereby make different predictions for sentences with the same direct object. This latter property leads BERT to correctly distinguish cases like *Nikolai finished a piece of stewed rabbit* (*eat*) from *Annie finishes the piece, lowering the bow* (*play*), where EBL predicts *see* in both cases, and the other models predict *take*. The fact that the use of contextualized, token-level representations leads to an increased performance on non-Blank responses suggests that information beyond direct-object noun is relevant in establishing the inferred event, where there is one (i.e., for the HC and LC cases).

Sentence	Responses	Predict top verb (§4; **-T**)	Allow for blanks (§5; **-B**)	Predict broad sense (§6; **-S**)
Lordier began the painting with a very light sketch...	BLANK (12), draw (2), redecorate, sketch, create, brush, paint	draw (11%)	BLANK (63%)	BLANK (63%)
Those so inclined can start the meal with vodka and tonic...	eat (7), BLANK (6), have (5), pair	eat (37%)	eat (37%)	CONSUME (63%)
She finished his back, then rearranged the sheet to do his legs. The top half of him was loose as a fish...	massage (11), BLANK (4), stretch (2), do (2), cover	massage (55%)	massage (55%)	OTHER (80%)
Although she was in residence for only about ten months she probably completed as many as ninety vases.	make (5), sculpt (3), BLANK (3), build, craft, create, shape	make (33%)	make (33%)	CREATE (73%)

Table 2: Example sentences, responses, and gold-standard answers for each dataset

5 Modelling blank responses

In a way, the approach taken in §4 set the models up to fail, as they have no mechanism to recognize that in cases such as (1-a) and (1-c), the interpretation does not 'need' to be resolved. In the remainder of this paper, we present two simple steps in the direction of broadening models' ability to interpret these cases in a way that is in line with their analysis as presented in §2. First, in this section, we update the models to explicitly predict the label to be 'blank' for items whose dominant response was blank. Then, in §6, we consider low-consensus items as cases where a broad sense is available for the event interpretation, but no specific verb resolution is necessary. These two strategies differ significantly from other datasets that approach this issue. Our changes in the nature of the correct label are illustrated in Table 2.

5.1 Updating models to predict 'blanks'

If we accept 'blanks' as valid modal participant responses to complement coercion interpretation cases, we need to allow models of complement coercion to have decision mechanisms to predict that response. For all unsupervised models, it is relatively straightforward to extend the unsupervised models by building in a threshold of confidence below which the models predict a blank response, reflecting the intuition that there is no single good verb the model can predict in response to the item. We call these models as a group **-B** ([B]lank predicting) models. As each model uses a different type of metric, applying a confidence threshold looks different for each one. We tuned specific thresholds by maximizing overall accuracy on our training set.

For the COOC model, for a given direct object and all uses of a verb with that direct object in the corpus, calculate the percentage p of all uses comprised by the top verb. If p is above a set threshold k, predict the top verb. Otherwise, predict a blank. We report results for an optimal k=12%. For the PROTO model, we build in a threshold based on the cosine similarity of the prototype vector to its nearest verb neighbor. If the similarity exceeds k, predict the verb corresponding to that nearest neighbor. Otherwise, predict a blank. (Optimal k=0.79). Finally, for BERT, we currently predict the verb with the highest confidence from BERT's masked prediction. To build in a threshold, we manually limit the number of predictions for the blank to check – if no verb is found in the top k words, predict a blank instead (Optimal k=5).

5.2 Results & Discussion

Table 1 presents the results for the blank-predicting models on the rows marked as **-B**. The addition of a tuned threshold mechanism to predict blanks improves accuracy on the Blanks subset (and thereby on the Overall accuracy) for all three models. For example, on the sentence *Some people with entrepreneurial spirit are still starting ___ farms*, all three blank-predicting models correctly predict a blank where their original versions incorrectly attempted verb responses.

However, this improvement comes at a cost for the COOC model (a drop from .439 to .293 for HC and .317 to .275 for LC items) and the PROTO model (a drop from .488 to .073 for HC and .250 to .100 for LC items). Looking at the model predictions, the tuned threshold leaves these two models with a very good recall for predicting Blank responses, but a low precision: both models mark many cases that have a dominant verb response as 'blanks'. For instance, in *The others are already finishing their granola* the dominant response *eat* is correctly predicted by the original COOC model, but the updated model wrongly predicts a blank. The decrease in performance on HC and LC cases is much smaller for BERT, with accuracies dropping only from .731 to .682 (HC) and from .425 to .400 (LC). However, BERT only predicts 43.5% of the Blank items correctly, compared to 71% (COOC) and 62.3% (PROTO) of cases.

What we take this to mean is that sentences with a dominant 'blank' response have a particular contextual profile: they may have different kinds of direct objects, or they may contain more adjunct phrases of the kind of *with a very light sketch* in example (1-a). Such properties could make models recognize comparably reliably that the interpretation should be a blank. (A further investigation of these contextual properties is left for future research.) Simple unsupervised models overgeneralize that blank-prediction, but for BERT the explicit prediction of blanks comes at a comparably low cost, suggesting that there is contextual signal that correlates with participant responses being dominantly 'blank'. We take this to be converging evidence for the coherence of a group of 'blank-dominant responses' as a distinct type of complement coercion responses.

6 Modelling low-consensus responses

We next consider expanding the interpretation of our models to better handle low-consensus cases. Just as we modelled 'blank' interpretations by expanding the possible predictions of models, we adapt our task to low-consensus cases by changing the possible predicted classes.

One practical problem with these cases is that the task of predicting a single verb might penalize a model which guesses a different but semantically very similar token from the correct top response. For example, in sentence (6), our BERT-B model incorrectly guessed *construct* instead of the top response *build* given by 7/19 participants. Although intuitively these answers both involve creation of an object through handiwork, our dataset judges one correct and one incorrect.

(6) Amtrak has recently announced that it will begin ___ a high-speed rail system connecting New York, Boston, and Washington, D.C., in 1998.

In the case of these low consensus sentences, predicting a single verb might be too restrictive. Instead, we consider a second change to the evaluation and models. Namely, we replace individual verb answers with broad senses of meaning that cover a shared property of multiple verb responses across items.

A few approaches have previously modelled the concept of broader senses in complement coercion interpretations. For example, Shutova and Teufel (2009) clustered many possible interpretations to short verb+object phrases. For the pair *finish video*, this includes *film, shoot, take, produce, make. . .* as one cluster, *watch, view, see, examine. . .* as another, and *edit, cut, redact, screen. . .* as a third. Models were then evaluated on how closely their unsupervised clustering of the same items matched annotator clusters. Our modelling differs from this unsupervised clustering in two key ways. First, we pre-define senses that apply across many coercion phrases, rather than creating clusters for specific verb+object combinations. Second, we test predictions on individual sentences rather than predicting one cluster over all possible sentences for ambiguous phrases.

6.1 Updating models to capture broad senses

Modeling responses with broad meaning senses requires two updates: (1) reworking our dataset where correct answers consist of broad senses, and (2) updating our models to be able to predict these new classes. Among the responses to our elicitation task, we found two coherent groups of broad senses of verbs: verbs that involve some form of creation (CREATE; e.g., *make, build, write*) and verbs involving some form of consumption (CONSUME; e.g., *read, eat, drink, watch*). All unique responses were manually assigned to one of these two groups, or tagged as OTHER (e.g., *destroy, massage, hold*) if they didn't belong to either group. For each sentence, we then define the "correct" broad sense as the most popular broad sense category among all participant responses to the sentence.

In order to update our models so that they predicted broad senses rather than individual verbs, we needed a procedure to map a model's single verb prediction into a broad sense category. We used the hand-labels of senses for the gold standard answers to automate the prediction of a broad sense based on a model's originally predicted verb. For each category, we combined word vectors for all tagged examples into a single averaged vector representing that sense. When a model would predict a single verb V, it instead predicts the category whose average vector is closest to the vector for V. In this way, we update the verb predicting models to instead predict broad sense labels.

6.2 Results & Discussion

We report performance of each model in Table 3. We used the same training/test split as in the original dataset, and kept all sentences where the dominant response was a 'blank.' As in §4 and §5, we report the accuracy overall and on the three subsets of the data. This means the gold-standard for our updated dataset includes only 4 possible classes to predict: the 3 broad senses CREATE, CONSUME, OTHER, as well as BLANK.

Given the different inventory of categories, a direct comparison with the results in §5 is not possible; instead we compare relative increases of performance across models and subsets of the data. Because the broad sense predictions are derived from the verb prediction of the same models, we do note that sentences where a -T or -B model predicted incorrectly but a -S model predicted correctly are cases where the model predicted the correct sense but failed to match the exact verb. Sentences where all models were incorrect are sentences where the prediction belongs to an entirely different event sense (e.g., predicting *cook*/CREATE when the gold standard is *eat*/CONSUME).

High-consensus vs low-consensus: For BERT and EBL, the two best performing models on the broad sense dataset, we remark that the HC and LC cases show similar accuracies. In contrast to other modelling in §4 and §5, where accuracy was low for low-consensus items in particular, the broad sense prediction task shows less difference between the categories.

This improvement for low-consensus cases can be attributed to low-consensus sentences which received incorrect single verb predictions on the previous task but now received the correct broad interpretation. For example, in sentence (7), BERT-B predicts *draw* which was incorrect for the verb prediction task (correct: *write*, 6/19 participants). However, *draw* is categorized under the sense CREATE for the broad sense task, which is correct with 11/19 responses.

	Overall	Blanks	HC	LC
EBL-S	.673	.870	.561	.450
COOC-S	.487	.710	.317	.275
PROTO-S	.393	.623	.219	.225
BERT-S	.547	.435	.634	.650

Table 3: Accuracy for 4 models on the modified broad-senses dataset as well as its three subsets.

(7) McGruder began ___ his politically charged hip-hop comic strip for his college newspaper

The close performance for HC and LC cases on the broad senses task supports our intuition that many coercion sentences involve broad sense interpretation. We suggest that individual verb paraphrases for interpretations may be an artefact of tasks that prompt specific verb responses. That is, participants may be able to provide specific verb interpretations when prompted, but outside of the context of the elicitation task only resolve the broad sense and leave the specific nature of the event unspecified.

7 Discussion

By reframing the possible classes predicted in a coercion interpretation task, we break from the typical paradigm of considering complement coercion as analogous to verb paraphrase, that is: as an obligatory (semantic) resolution of a particular event. In redefining the computational task we also reconsider how complement coercion has traditionally been represented – as a type-shift from object to event, or a pragmatic process of interpreting a relevant scale. We acknowledge that neither of these accounts is formulated for modeling interpretation as a predictive task, and as such our work does not constitute a full comparison of all aspects of these accounts. Nonetheless, the fit with these accounts differs for the property at issue in our work, namely whether event resolution is obligatory.

The type-shifting account of Pustejovsky and

Bouillon (1995) forwards that each direct object contains within its lexical-semantic qualia the possible event interpretations for a coercion sentence. This translates readily to a single-verb prediction task. Under a generous reading, we could broaden this account to explain cases like (6) where multiple semantically similar verbs (e.g., *construct, build*) make valid interpretations. That is, we can rethink the type-shift accounts to consider broad event senses rather than specific paraphrases, just as we did for low-consensus cases in §6.

Still, the prevalence of sentences where a blank response was dominant suggests there are many coercion sentences where a verb paraphrase is difficult to access or absent altogether. Building models that predict blanks is difficult to link to a type-shifting account, which builds on the assumption of complement coercion as a process of event interpretation based on the direct object noun. In contrast these examples are well covered by the scalar interpretation account of Piñango and Deo (2016) which does not necessitate an event interpretation.

Although our work is not intended as a model of the psycholinguistic result that complement coercion incurs processing cost, we remark that most experimental work has used stimuli which are deliberately high-consensus constructed examples. Our present work illustrates the breadth of complement coercion sentences and outlines three general patterns – high-consensus, low-consensus, and majority blank responses – only the first of which is represented in experimental stimuli.

Noteably, Frisson and McElree (2008) investigate the effect of response consensus on processing cost, finding no difference in cost for reading sentences with high vs. low consensus. This finding is used to show that ambiguity between interpretations does not modulate the processing cost. Our introduction of low-consensus cases which share a broad sense complicates this picture by suggesting that not all low-consensus coercion sentences involve ambiguity between broad senses. As well, even norming work from Frisson and McElree (2008) forced participants to choose a verb, leaving potential blank cases unexplored. While our findings do not make predictions for the processing cost observed in experimental work, they suggest potential new classes of experimental stimuli for future work in the form of low-consensus items with a single broad sense and blank-dominant cases.

Overall, the broad spectrum of coercion examples covering multiple sub-classes illustrates that the process of interpretation goes beyond selecting a single appropriate verb paraphrase. Indeed, the presence of many blank responses suggests that it may go beyond event interpretation as well. As such, our work suggests that using naturalistic data and analyzing the semantic-pragmatic properties of observed cases is critical to developing a more complete insight into a phenomenon like complement coercion.

References

Emmanuele Chersoni, Alessandro Lenci, and Philippe Blache. 2017. Logical metonymy in a distributional model of sentence comprehension. In *Sixth Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 168–177.

Mark Davies. 2009. The 385+ million word corpus of contemporary american english (1990–2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics*, 14(2):159–190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Yanai Elazar, Victoria Basmov, Shauli Ravfogel, Yoav Goldberg, and Reut Tsarfaty. 2020. The extraordinary failure of complement coercion crowdsourcing. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 106–116.

Steven Frisson. 2009. Semantic underspecification in language processing. *Language and Linguistics Compass*, 3(1):111–127.

Steven Frisson and Brian McElree. 2008. Complement coercion is not modulated by competition: Evidence from eye movements. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 34(1):1.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Maria Lapata and Alex Lascarides. 2003. A probabilistic account of logical metonymy. *Computational Linguistics*, 29(2):261–315.

Alessandro Lenci. 2011. Composing and updating verb argument expectations: A distributional semantic model. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 58–66.

Brian McElree, Matthew J Traxler, Martin J Pickering, Rachel E Seely, and Ray Jackendoff. 2001. Reading time evidence for enriched composition. *Cognition*, 78(1):B17–B25.

Stephen McGregor, Elisabetta Jezek, Matthew Purver, and Geraint Wiggins. 2017. A geometric method for detecting semantic coercion. In *IWCS 2017-12th International Conference on Computational Semantics-Long papers*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Maria Mercedes Piñango and Ashwini Deo. 2016. Reanalyzing the complement coercion effect through a generalized lexical semantics for aspectual verbs. *Journal of Semantics*, 33(2):359–408.

James Pustejovsky and Pierrette Bouillon. 1995. Aspectual coercion and logical polysemy. *Journal of Semantics*, 12(2):133–162.

Kirk Roberts and Sanda Harabagiu. 2011. Unsupervised learning of selectional restrictions and detection of argument coercions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 980–990.

Ekaterina Shutova and Simone Teufel. 2009. Logical metonymy: Discovering classes of meanings. In *Proceedings of the CogSci Workshop on Semantic Space Models*, pages 29–34. Citeseer.

Testable. testable.org: One-stop solution for behavioral experiments, surveys, and data collection.

Matthew J Traxler, Martin J Pickering, and Brian McElree. 2002. Coercion in sentence processing: Evidence from eye-movements and self-paced reading. *Journal of Memory and Language*, 47(4):530–547.

Cornelia Maria Verspoor. 1997. Conventionality-governed logical metonymy. In *Proceedings of the Second International Workshop on Computational Semantics*, pages 300–312. Citeseer.

Alessandra Zarcone and Sebastian Padó. 2010. “i like work: I can sit and look at it for hours” type clash vs. plausibility in covert event recovery. *Proceedings of Verb 2010*, page 209.

Alessandra Zarcone, Jason Utt, and Sebastian Padó. 2012. Modeling covert event retrieval in logical metonymy: probabilistic and distributional accounts. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*, pages 70–79.

A Appendix - Corpus Extraction Heuristics

For extracting likely complement coercion candidates in §2, we used a series of heuristics to narrow from corpus sentences to coercion candidates.

First, we extracted all sentences which used one of five aspectual verbs (*begin, complete, end, finish, start*) if the sentence also included an overt direct object. We then eliminated uses which included an overt complement verb (*I had just finished washing the dishes*). This left 44,810 aspectual verb sentences from the corpus. We further removed sentences at the beginning or end of a passage in the corpus, i.e., sentences where we could not present at least one other sentence of context on either side. This left 41,372 aspectual verb sentences.

Next, we used information about the direct object in the ontological database WordNet to remove non-coercion uses of aspectual verbs (Miller, 1995). While we suspect there is common ground between many aspectual verb uses regardless of direct object type, other work leaves out specific direct objects as separate senses (Verspoor, 1997; Elazar et al., 2020). We choose to narrow the field here for maximum analogy to past work. Specifically, we removed all sentences where the direct object had no extension which was a physical entity. This included unclear cases where a physical meaning was possible but not certain – for example, "work," "school," or "company" and any direct object ending in *-ion*, those being event nouns. We also removed sentences where the aspectual verb *start* took a direct object with a "motor vehicle" sense in WordNet, (e.g., *...start the car/engine*). This left 5,088 candidate coercion sentences.

Finally, we removed sentences with a particle, e.g., (*finish* up *the tea*). While these sentences do resemble complement coercion in most respects, we expected they would introduce grammaticality issues with the fill-in-the-blank paraphrase task, potentially discouraging paraphrases when a clear interpretation was available.

This process left 4,583 sentences where an aspectual verb takes a clear entity object, resembling complement coercion by all definitions in the literature. Of these, we randomly selected 300 to use for crowdsourcing.

B Appendix - Dataset Materials

B.1 Materials

For our crowdsourcing experiment, we recruited online via Testable, under approval from **Anonymous Institution**. Participants were paid at a rate of $15CAD per hour, for annotating 50 items taking approximately 20 minutes.

Participants were initially given the following instructions asking them either provide a verb paraphrase or leave a blank (boldface as presented to participants):

> In a sentence like "The thirsty athlete finished a bottle of water," we know that the athlete drank the bottle of water, even though the verb "drinking" is not present. We are interested in sentences where such "silent verbs" are and aren't present.
>
> In this survey, you will be shown sentences which may or may not have this kind of silent verb meaning. We have added a blank line to the sentence where a verb might go if available. You will have the option to "**fill in the blank**" to make the meaning explicit or characterize the event occurring. For example, given sentence (A)...
>
> (A) The thirsty athlete finished ___ a bottle of water.
>
> ...you might choose to fill in the blank with "drinking" Given sentence (B)...
>
> (B) The construction company completed ___ a new condo.
>
> ...you might choose to fill in the blank with "building" **If you choose to fill in the blank, please fill it with a single verb with an "-ing" ending.** Some sentences might not have any reasonable input. In these cases, **you may leave the input blank.** For example, in (C)...
>
> (C) I began ___ the day by stretching.
>
> ...the sentence is fine as is, and doesn't necessarily imply a specific verb. You might choose to leave the sentence blank if you cannot think of any reasonable verb. Don't spend too much time on any one item – your gut feeling is most important. If you don't think of any verb after a few seconds, leave it blank and move on to the next question.

Items were presented in groups of 5. Participants were reminded of the instructions after every block of 5 items. An example of 5 items as displayed in a browser is shown in Figure 1.

B.2 Example items and responses

In this section we include 3 item examples from each of the 3 categories discussed in §2: high-consensus (top response above median consensus), low-consensus (top response below median consensus), or blank (top response was a blank).

B.2.1 High-consensus examples

(8) All this attention The Third Policeman is getting would've stunned its author. He finished ___ the book in 1940 at the dawn of World War II. Bad timing for a comic novel.
TOP RESPONSE: *write*, 100% (15/15)
ALL RESPONSES: writing (15)

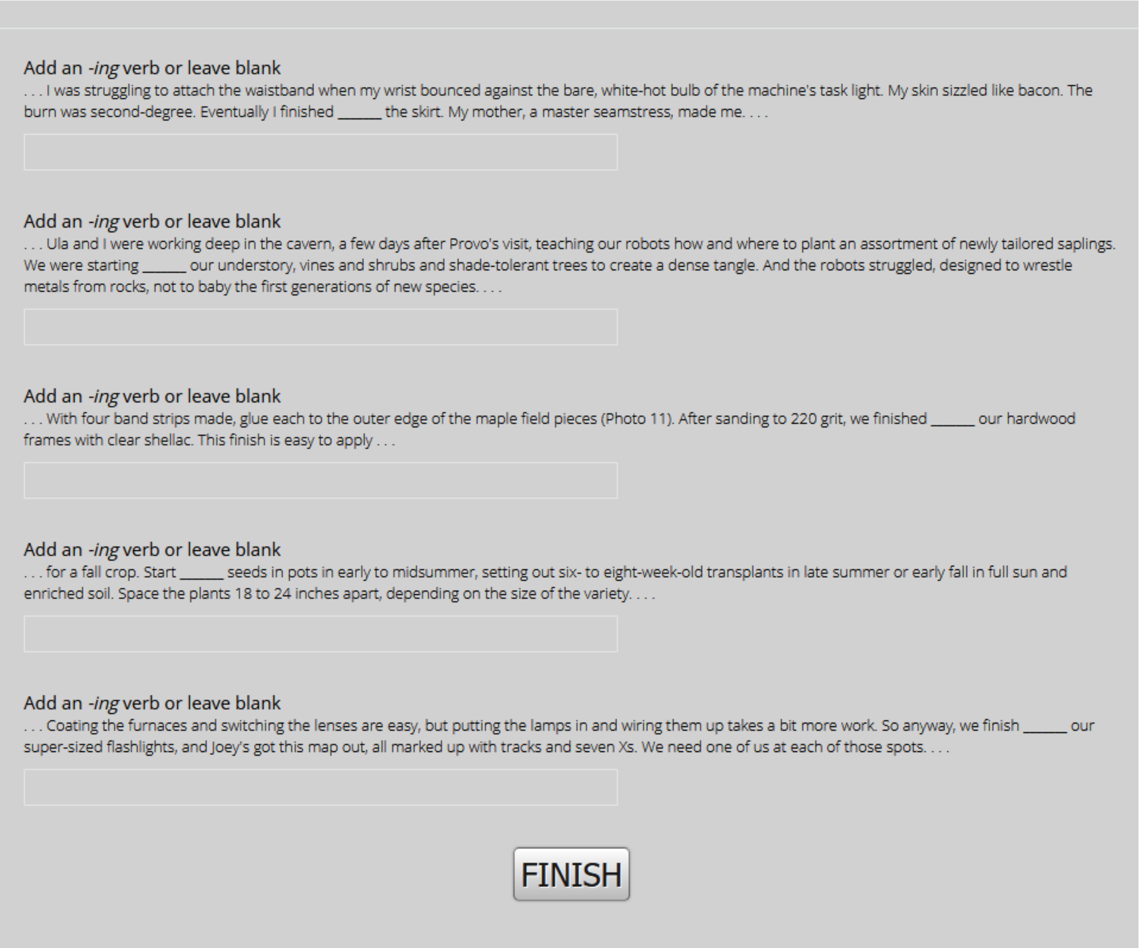
(9) Yet a closer look reveals subtle touches of Sikes' brush. He finished ___ the walls in an aged plaster texture in warm shades of light gold and gray. He marbleized a pair of columns in similar neutral tones yet made them pop with metallic gold accents.
TOP RESPONSE: *paint*, 93.3% (14/15)
ALL RESPONSES: painting (14), building (1)

(10) Have I done something before 9/11 or after? When did I start ___ guitar? That was after.
TOP RESPONSE: *play*, 73.7% (14/19)
ALL RESPONSES: playing (14), learning (4), practicing (1)

B.3 Majority blank examples

(11) The peace has proven lasting, much to WIPNET's surprise. Gradually Liberia has started ___ the long road back from war. The country was absolutely devastated by 13 years of war, says Bushkofsky.
TOP RESPONSE: *BLANK*, 70.0% (14/20)
ALL RESPONSES: BLANK (14), taking (2), recovering (2), walking (1), building (1)

(12) Ants far exceed human beings in nastiness, Wilson has written. If ants had nuclear



Add an *-ing* verb or leave blank

. . . I was struggling to attach the waistband when my wrist bounced against the bare, white-hot bulb of the machine's task light. My skin sizzled like bacon. The burn was second-degree. Eventually I finished ______ the skirt. My mother, a master seamstress, made me. . . .

Add an *-ing* verb or leave blank

. . . Ula and I were working deep in the cavern, a few days after Provo's visit, teaching our robots how and where to plant an assortment of newly tailored saplings. We were starting ______ our understory, vines and shrubs and shade-tolerant trees to create a dense tangle. And the robots struggled, designed to wrestle metals from rocks, not to baby the first generations of new species. . . .

Add an *-ing* verb or leave blank

. . . With four band strips made, glue each to the outer edge of the maple field pieces (Photo 11). After sanding to 220 grit, we finished ______ our hardwood frames with clear shellac. This finish is easy to apply . . .

Add an *-ing* verb or leave blank

. . . for a fall crop. Start ______ seeds in pots in early to midsummer, setting out six- to eight-week-old transplants in late summer or early fall in full sun and enriched soil. Space the plants 18 to 24 inches apart, depending on the size of the variety. . . .

Add an *-ing* verb or leave blank

. . . Coating the furnaces and switching the lenses are easy, but putting the lamps in and wiring them up takes a bit more work. So anyway, we finish ______ our super-sized flashlights, and Joey's got this map out, all marked up with tracks and seven Xs. We need one of us at each of those spots. . . .

Figure 1: Screenshot of 5 items as displayed to a participant during annotation.

weapons, they would probably end ___ the world in a week. He told me there were only 20 people in the world who knew enough to identify and classify ants...
TOP RESPONSE: *BLANK*, 65.0% (13/20)
ALL RESPONSES: BLANK (13), destroying(4), annihilating(1), fighting(1), living(1)

(13) ...is the similar-size Keystone of Hercules. We complete our ___ circuit around the rim of the sky by looking southwest. Here dramatic Scorpius is well past its prime height, but it's still not too late for good looks at twinkling Antares and other illustrious Scorpius treasures.
TOP RESPONSE: *BLANK*, 65.0% (14/20)
ALL RESPONSES: BLANK (14), building (2), developing (1), doing (1), making (1), walking (1)

B.4 Low-consensus examples

(14) Erica became unconscious immediately. The technicians completed ___ the X-ray. Despite the Portlock's concerns, the technicians told them it was okay to take Erica home, even though she was still unconscious.
TOP RESPONSE: *take*, 30.0% (6/20)
ALL RESPONSES: taking (6), BLANK (5), analyzing (3), scanning (2), examining (1), making (1), performing (1), running (1)

(15) "He must like you a lot." Tyla finished ___ Julienne's hair, went to pick up her half-boots, and knelt to put them on her. Julienne was smiling, a dreamy, private softening of her lips.
TOP RESPONSE: *braid*, 21.1% (4/19)
ALL RESPONSES: braiding (4), combing (4), tying (3), brushing (2), BLANK (1), cutting (1), doing (1), making (1), styling (1), weaving (1)

(16) Painting outdoors allows me to capture light and color with much greater accuracy, he notes. He may complete ___ a piece during his first outing or return to the location the next day, but he often finishes paintings in his studio, as he did with Carmel Mission Bell Tower. Durborow especially enjoys the friendly competition and camaraderie of paint-outs, where artists work together on location, and he attends at least four such events a year.
TOP RESPONSE: *paint*, 33.3% (5/15)
ALL RESPONSES: painting (5), BLANK (3), drawing (3), assembling (1), building (1), making (1), writing (1)

Representing multiple dependencies in prosodic structures

Kristine M. Yu
University of Massachusetts Amherst / Amherst, Massachusetts, USA
krisyu@linguist.umass.edu

Abstract

Association of tones to prosodic trees was introduced in Pierrehumbert and Beckman (1988). This included: (i) tonal association to higher-level prosodic nodes such as intonational phrases, and (ii) multiple association of a tone to a higher-level prosodic node in addition to a tone bearing unit such as a syllable. Since then, these concepts have been broadly assumed in intonational phonology without much comment, even though Pierrehumbert and Beckman (1988)'s stipulation that tones associated to higher-level prosodic nodes are peripherally realized does not fit all the empirical data. We show that peripherally-realized tones associated to prosodic nodes can be naturally represented with bottom-up tree transducers. Additionally, multi bottom-up tree transducers provide a way to represent non-peripheral boundary tones and multiple tonal association, as well as multiple dependencies in prosodic structures in general, including prosodically-conditioned segmental allophony.

1 Introduction

It is widely accepted that describing segmental and tonal distributions and processes over trees built with prosodic constituents (e.g., syllables (σ), feet (Ft), prosodic words (ω), accentual phrases (α), phonological phrases (φ), and intonational phrases (ι)) can help capture phonological generalizations. A classic example exemplifying this comes from Bengali (Hayes and Lahiri, 1991). As exemplified in Fig. 1, adapted from Khan (2008, p. 101)[1], rises in the pitch contour delineate phonological chunks in Bengali. In the example in Fig. 1, these chunks happen to be the size of a morphosyntactic word plus affixes, but chunk size can vary depending on speech rate. For example, Hayes and Lahiri (1991, (54)) provides the example in (1), where we indicate phonological chunks delineated by melodic rises using square brackets. In (1a), one melodic rise occurs per word as in Fig. 1. However, at faster speech rates, a speaker may utter the same sentence with the prosodic chunkings in (1b) or (1c), or at an even faster speech rate, as (1d).

(1) Variation in prosodic domains

a. [ɔmor] [t͡ʃador] [tara-ke] [diet͡ʃʰe]
Amor scarf Tara-obj gave
'Amor *gave* a scarf to Tara'

b. [ɔmot̠͡ʃ̠ t͡ʃador] [tara-ke] [diet͡ʃʰe]

c. [ɔmor] [t͡ʃadot̠ tara-ke] [diet͡ʃʰe]

d. [ɔmot̠͡ʃ̠ t͡ʃadot̠ tara-ke] [diet͡ʃʰe]

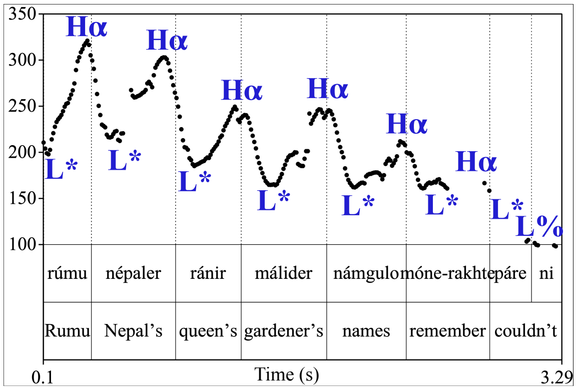


Figure 1: Melodic rises in Bengali analyzed as tonal sequences. Fundamental frequency (Hz) on y-axis, time (s) on x-axis. 'Rumu couldn't remember the names of the gardeners of the queen of Nepal.' Example from Khan (2008, p. 101).

It would be difficult to characterize all of these possible chunkings of the same sentence under the same information structural conditions as being

[1] The Bengali case study presented in this paper is based on Hayes and Lahiri (1991)'s analysis of a Kolkata variety, but we show a pitch track example from Khan (2008)'s analysis of a Bangladeshi variety since recordings from Khan (2008) are readily available.

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 171-183.
Held on-line February 7-9, 2022

morphosyntactically-conditioned. Moreover, the same chunks delineated by melodic rises also determine whether two other segmental processes occur (Hayes and Lahiri, 1991, §§9.1, 9.2): (i) total assimilation of /r/ to an immediately following coronal consonant, and (ii) voicing assimilation of a stop to an immediately following stop. These two segmental processes occur when both the segment that gets changed as well as its conditioning environment occur within the same chunk, as exemplified for the final [r]s in [ɔmor] and [t͡ʃador] in (1), which are underlined when they assimilate to [t͡ʃ] and [t], respectively. (Note: Hayes and Lahiri (1991) calls these prosodic constituents phonological phrases, while Khan (2008) calls them accentual phrases; here we use 'accentual phrase').

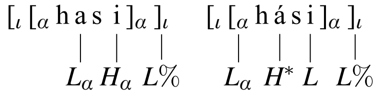
The generalization that melodic patterns delineate the edges of prosodic constituents also motivated one of the foundational assumptions of Autosegmental-Metrical (AM) Theory (Pierrehumbert, 1980; Pierrehumbert and Beckman, 1988; Ladd, 1996; Arvaniti and Fletcher, 2020), a theory that dominates work on intonational phonology: the assumption that tones can associate not only to tone-bearing units (TBUs) "at the bottom of the tree" (i.e., non-terminal nodes that immediately dominate terminals) such as moras (μ) and syllables (σ), but also to any higher-level node in the prosodic tree, e.g., the accentual phrase or the intonational phrase (Pierrehumbert and Beckman, 1988, p. 21). While the concept of tones associating to TBUs was carried over from Autosegmental Theory (Goldsmith, 1976), the concept of tones associating to prosodic constituents in general was an innovation of AM theory, as well as the notion that tones can be multiply associated—both to a higher-level prosodic node as well as a TBU (Pierrehumbert and Beckman, 1988).

In Fig. 1, each melodic rise is analyzed as the phonetic realization of a sequence of two discrete tones: a low pitch accent (L*), and a high accentual phrase tone (H_α). The '*' diacritic indicates a pitch accent; the 'α' diacritic indicates an accentual phrase tone. The entire sentence comprises an intonational phrase, with a low intonational phrase tone, L%, at the right edge (the '%' diacritic indicates an intonational phrase tone). In AM Theory, a pitch accent like L* is a tone whose appearance and temporal location are determined by accented TBUs, i.e., TBUs with "an abstract phonological location indicator of tone" (Gussenhoven, To appear, §1.2) and is represented as being associated to an accented TBU. An edge tone like H_α or L% is a tone whose appearance and location is determined by prosodic constituent edges and is represented as being associated to a prosodic node at a higher-level node than the TBU.

The L* appears at the left edge of an accentual phrase, while the H_α appears at the right edge of an accentual phrase. So why is the L defined as a pitch accent rather than an edge tone? In Bengali, accented TBUs are syllables that receive stress, and Bengali has word-initial stress—thus, the L tones are always word-initial in Fig. 1. However, Hayes and Lahiri (1991, p. 56) shows that when a word is preceded by a clitic, the L tone is not phrase-initial and appears instead on the initial syllable of the word, after the clitic—thus tracking the accented TBU rather than the left edge of accentual phrases.

The Bengali example in Fig. 1 exemplifies the distinction between pitch accents and edge tones, but what about the concept of the association of a single tone to both a TBU as well as a higher-level prosodic node? Multiple association of this kind was first motivated by Pierrehumbert and Beckman (1988) for Tokyo Japanese due to differences in the phonetic realization of L_α tones systematically conditioned by the position of lexical accent. In Japanese, accented syllables are lexically specified and receive a bitonal H*+L tone, cf. *hasi* 'edge' vs. *hási* 'chopsticks' vs. *hasí* 'bridge' (Gussenhoven, 2004, p. 186), where accent is indicated with an acute accent mark. The comparison between unaccented *hasi* and initially-accented *hási* is represented in (2) using association of tones to labeled brackets for singly associated accentual phrase and intonational phrase tones, following notational conventions popularized by Hayes and Lahiri (1991). The analysis shown follows Gussenhoven (2004, 2014).

(2) Tonal associations for *hasi* vs. *hási*



In words like *hási* where a tone occupies the first TBU, i.e., the first mora, the word-initial L_α is pronounced with a mid pitch, but in words like *hasi* where no lexical accent occupies the first TBU, the L_α is pronounced fully low, see Pierrehumbert and Beckman (1988, §5.5); Gussenhoven (2004, p. 189). This difference is attributed to a difference in association: in *hasi*, the first TBU is available for the L_α to associate and so it associates not only to the α node but also this TBU; in *hási*, the first TBU is unavailable so the L_α is associated only to the α node. Similarly, phonetic evidence shows that the L of the lexical accent associates to an unoccupied TBU immediately following the accented TBU (Gussenhoven, 2014, §2), as shown for *hási* in (2). There is also an H_α following the peripheral (i.e., at the left edge) L_α. In *hasi*, the second TBU is available for the H_α to associate to, but since the second TBU of *hási* is occupied by the L of the lexical accent, the H_α is deleted. Non-peripheral, unassociated tones are deleted in Japanese (Gussenhoven, 2014, §2).

The concepts of association of tones to higher-level prosodic nodes and multiple tonal association introduced in Pierrehumbert and Beckman (1988) have been broadly assumed in intonational phonology without much comment (but see, e.g., Prieto et al. (2005); Gussenhoven (2018) for exceptions). However, while the computational properties of association of tones to TBUs have received much attention, e.g., Chandlee and Jardine (2021) and references therein, the computational properties of tones associating to prosodic trees, i.e., tones as terminals participating in dominance relations in prosodic trees, as well as multiple tonal association to TBUs and prosodic nodes, have not. In fact, as noted in Pierrehumbert (2011, p. 5), prosodic trees with multiple tonal associations are technically not trees anymore, since terminal nodes can have more than one parent.

Moreover, the formal properties of tones associating to prosodic trees defined in Pierrehumbert and Beckman (1988, Ch. 6) have not been revisited, although Pierrehumbert and Beckman (1988, Ch. 6) stipulates the temporal location of tones associated to prosodic nodes (i.e., edge tones, or boundary tones) to be at the periphery of the constituent they are associated to. The stipulation is problematic because Gussenhoven (2000) provides examples from Roermond Dutch where edge tones are not peripherally realized, i.e., a lexical accent tone is sequenced to appear after a right-edge aligned intonational phrase boundary tone. Gussenhoven (2000)'s response to the problematic peripherality stipulation (see also Gussenhoven (2018, §4)) is to abandon the idea of tonal association to higher-level prosodic nodes altogether in favor of Align constraints between tones and prosodic constituents. But the theory of tonal association to higher-level prosodic nodes as proposed in Pierrehumbert and Beckman (1988, Ch. 6) has remained a fundamental assumption of Autosegmental-Metrical Theory (Arvaniti and Fletcher, 2020), despite its inability to allow for non-peripheral prosodic boundary tones.

This paper shows that standard tools from formal language theory can be used to formalize the notion of tonal association to prosodic trees and handle both multiple tonal association and non-peripheral boundary tones. To define tonal association in prosodic trees, we make use of finite state tree rewrite grammars, which can be recognized by bottom-up tree transducers (Baker, 1978; Comon et al., 2007), and in the paper, we use the notation of finite state tree transducers to define our tree grammars (Rounds, 1970). The bottom-up tree transductions provide a natural mechanism for prosodic boundary tones to be sequenced peripherally, without stipulation.

Moreover, we show that a standard extension of bottom-up tree transducers—multi bottom-up tree transducers (mbutts) (Lilin, 1978; Fülöp et al., 2004; Maletti, 2008), see Maletti (2008, §4) for a formal definition—can represent multiple tonal association and allow non-peripheral edge tones. String yields from trees that can be built with finite state bottom-up tree transducers are context-free, i.e., strings that can be derived with CFG grammars (Comon et al., 2007, §2.4). String yields from trees that can be built with multi finite state bottom-up tree transducers are strings that can be derived with multiple CFGs (Engelfriet et al., 2009), grammars that that are more expressive than CFGs, in which one constituent can enter into relationships with two of its ancestors, e.g., in syntactic movement, see Clark (2014).

While mbutts have been used to express syntactic relations (Kobele et al., 2007; Graf, 2012) and also syntax-prosody mapping (Dolatian et al., 2021), we show here—building on Yu (2021)—that mbutts are of interest as representations for phonological phenomena in general. Multiple tonal association is only one instance of multiple dependencies in prosodic trees, but we show that so are prosodically-conditioned segmental processes such as Bengali r-assimilation, and that mbutts can handle these processes as well. The next section, §2, introduces a first tree transduction for single tonal associations in a single word of Bengali. §3 introduces mbutts in tree transductions for tone association in Japanese for *hasi* and *hási* in (2), and §4 shows how mbutts can represent r-assimilation in Bengali, too. §5 discusses issues raised by using mbutt representations.

2 A first tree transduction

A finite state bottom-up tree transducer can be thought of as a generalization of a string finite state transducer that can process multiple branches rather than a single branch (a string). A string finite state transducer processes a string from left to right, one symbol at a time, and enters one of finitely many states after each step. A string transduction is recognized as well-formed if and only if the transducer enters a final state after processing the entire string. A finite state bottom-up tree transducer processes a tree from leaves towards the root, one subtree at a time, and enters one of finitely many states after each step. A tree transduction is recognized as well-formed if and only if the transducer enters a final state after processing the tree all the way up to the root. A tree transduction step can re-label nodes, delete subtrees, or insert new material. However, bottom-up tree transductions cannot change structures that have already been built.

As a first introduction to tree transductions, we show the grammar and steps to insert the pitch accent (L*) and accentual phrase tone (H_α) and assign stress in an accentual (α) phrase of a single two-syllable prosodic word (ω) in Bengali, e.g., /t͡ʃador/ or /ɔmor/ from (1). (An even simpler warm-up transduction that inserts just the H_α and ignores stress and pitch accent assignment is given in Table 5 and (7) in Appendix A.) For this first transduction, we make the simplification that the pitch accent insertion rule in Bengali is only ω-based, i.e., a pitch accent is assigned to the stressed syllable in each ω. A transduction that assigns an L* to the stressed syllable of only an α-initial ω is shown in Appendix B.

Since the segments play no role in these processes, we leave them out and only show tonal association to the syllabic TBUs (σ) and α node. The rules in (3) take the input tree shown as the leftmost tree in the derivation in Table 1 and returns the rightmost tree in Table 1 as the output tree (ignoring the green filled circle at the moment). We assume a lexicon of low and high tones and a placeholder symbol, ε, that indicates a location where a tone can be filled, $\{L, H, \varepsilon\}$, and we define q_α to be a final state. A green filled circle decorating a tree in Table 1 indicates which state the transducer enters after the application of the transition rule labeling the rewrite arrow to the left of the tree, and the output at each step is shown as the subtree under the state. By convention, a state is positioned as the mother node of the subtree that has just been processed, but isn't actually part of the tree—it's just an annotation like a "you are here" marker.

(3) Grammar fragment for tree transduction of single-ω accentual phrase; q_α final state

$$\begin{array}{ll}
[B1] & \varepsilon() \rightarrow q_\varepsilon(\varepsilon()) \\
[B2] & \sigma(q_\varepsilon(t)) \rightarrow q_\zeta(t) \\
[B3] & \omega(q_\zeta(t_1), q_\zeta(t_2)) \rightarrow q_\omega(\omega(\mathrm{str}(\sigma(L)), \sigma(t_2))) \\
[B4] & \alpha(q_\omega(t)) \rightarrow q_\alpha(\alpha(t, H))
\end{array}$$

The left-hand side of a rule shows the structure required for the rule to be applied, and its format differs depending on whether the transducer is at a leaf or not. When the transducer is at a leaf, e.g., Rule [B1], the left-hand side of the rule is just the leaf, which by definition, has no daughters underneath—indicated by the empty parentheses following the leaf label, e.g., $\varepsilon()$ in Rule [B1]. If the transducer is at an ε leaf, then Rule [B1] can apply, as shown in the first step in Table 1. The right-hand side shows the state entered, as well as the output, shown in the immediately following parentheses. For example, when the transducer applies Rule [B1], it processes the leaf $\varepsilon()$, enters state q_ε, and returns the input leaf $\varepsilon()$ unaltered, as output. And the first step in Table 1 shows the transducer processing both ε leaves with Rule [B1] (which is shown as applying twice with the notation $[B1]^2$) to enter state q_ε on both the left and right branches and output back ε on each branch, which is shown as the daughter of the q_ε node. The green circles in the derivation move from the leaves towards the root over the course of the derivation since the tree is processed bottom-up. When the transducer is at a non-terminal node (all rules but Rule [B1]), the current node label and the state(s) that the transducer is in must match the left-hand side of a rule for the rule to apply.

Rule [B2] states that if the transducer is at a unary σ node with its single daughter (variable t) in state q_ε, then the transducer can enter q_ζ, deleting the σ node but leaving its daughter (t) unchanged. The second step in Table 1 shows the transducer applying this rule for both the left and right branches; here, t is ε. Rule [B3] is a merge rule that states that if the transducer is at a binary-branching ω node

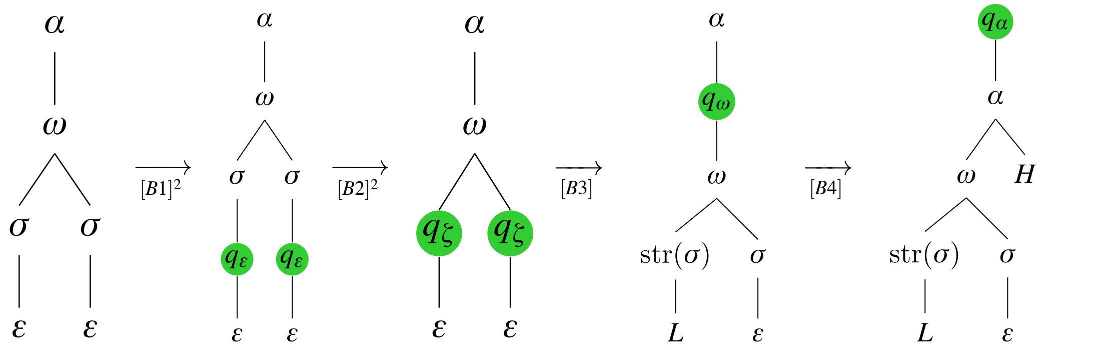


Table 1: Transduction of tone insertion and stress assignment in single-word accentual phrase using rules in (3)

with its left daughter (t_1) in state q_ζ and its right daughter (t_2) also in state q_ζ, then the transducer can enter q_ω and output back the ω subtree with σ nodes inserted above both daughters. Moreover, the σ dominating the first daughter (t_1) is assigned stress, $\mathrm{str}(\sigma)$,[2] following Bengali's ω-initial stress assignment rule, and its associated ε tone is replaced with a L tone, i.e., an L* pitch accent. Since the L pitch accent is already defined by where it is associated in the tree, there is no need to also add a '*' diacritic. The third step in Table 1 shows the transducer applying Rule [B3]. The replacement of ε with L in Rule [B3] is why Rule [B2] is defined to delete the σ node. The bottom-up transducer can modify the daughter tone of a σ node only if the σ node has not already been built. The L* associates to the stressed syllable, which is defined to be the ω-initial syllable. So stress assignment, and consequently pitch accent assignment, can only occur when the ω node is processed.

The transduction of the input tree can end successfully if the transducer completes processing the tree up to the root node and enters a final state—a state where the derivation can optionally terminate. Rule [B4] states that if the transducer is at an α node with a single daughter (t) in state q_ω, then the transducer can enter q_α and output back the α subtree with its daughter (t) unaltered and insert a new daughter H accentual phrase tone to the right. No tonal α diacritic is needed since the H tone is defined by where it associates in the tree. For the purposes of processing just an accentual phrase, we designate q_α as a final state.

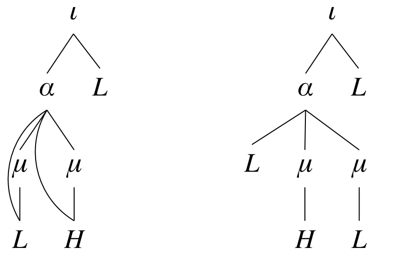
Upon the application of Rule [B4], the transducer has processed the entire tree up to the root, enters final state q_α (positioned as the mother node of the root node), and returns the output tree, which is shown as the daughter of q_α. Thus, the tree grammar in (3) recognizes that the transduction in Table 1 is well-formed. In fact, the transduction in Table 1 is the only transduction that (3) recognizes as well-formed. For instance, an output tree like the rightmost tree in Table 1, but with the second syllable under ω stressed rather than the first, is not well-formed under (3).

The rule to insert an accentual phrase tone in Table 1, Rule [B4], exemplifies how peripherality of tones associated to higher-level prosodic nodes is a natural consequence of the definition of bottom-up tree transducers. Since a bottom-up tree transducer cannot make changes to a subtree that has already been built, no rule in place of Rule [B4] can be defined to insert a tone inside the already-built ω-subtree. Rather, Rule [B4] inserts a tonal daughter of α that is a sister to the right of the already-built ω-subtree. Another possible rule in place of Rule [B4] could insert an H tone to the left of the ω-subtree, e.g., by replacing the right-hand side of Rule [B4], $q_\alpha(\alpha(t,H))$, with $q_\alpha(\alpha(H,t))$. The possible placements of an inserted tone are confined to the periphery of the accentual phrase. Non-peripheral boundary tones can be defined with multi bottom-up tree transductions, which we introduce in the next section.

3 Multiple dependencies in tonal associations

While the association of L* pitch accent in Table 1 is determined at the word-level, it does not have a multiple dependency because the L is inserted only at the step when the ω is processed (Rule [B3])—the L is not carried up the derivation over multi-

[2] We indicate a stressed syllable with $\mathrm{str}(\sigma)$ rather than a diacritic $\acute{\sigma}$ to make it explicit that the σ in $\mathrm{str}(\sigma)$ is copied and that a stressed σ isn't just another symbol with arbitrary relation to σ.



(a) Unaccented *hasi* (b) Accented *hási*

Figure 2: Derived trees showing tonal associations for *hasi* and *hási* described in (2).

ple steps. Multiple dependencies do, however, occur in the tonal associations of two-syllable words in Japanese such as *hasi* and *hási* (2), and mbutts give us a way to express them, as we show in this section. Since the transductions here involve only unary (and no bimoraic) syllables and tonal insertion is not conditioned on the prosodic word, we omit those constituents in the prosodic trees to conserve space. The final output trees to be derived, following (2), are shown in Figure 2. As is done in AM Theory as well as similar syntactic derivations, we explicitly indicate the multiple associations of the L and H tones in *hasi* by showing them each as having two mothers—the mora and the accentual phrase node. These kind of structures, where a single terminal node has two parents—can be interpreted as multidominance structures (Gärtner, 2002). (The multiple dependencies of the tones in accented *hási* are not explicitly represented in the same way in Figure 2 because they occur only in the course of the derivation and not also in the final output derived tree like for *hasi*.)

A transduction for tone assignment in an intonational phrase consisting of an unaccented 2σ word, e.g., *hasi*, is given in Table 2 and (4), and a transduction for tone assignment in *hási* is given in Table 3 and (5). State labels in common with those for the Bengali in §2 shouldn't be taken to identify shared states between the transductions.

The transduction for /hasi/ (Table 2, using the rules in (4)) must define L and H accentual phrase tones, meaning that L and H tones are to be inserted only once the α node is processed, as daughters of α. But these tones are also to be daughters of μ's, which themselves are daughters of α, and there cannot be any modifications to subtrees already built under the α node. Thus, much like in Table 1, after the ε leaves are processed without change (Rule [J1a]), the μ nodes are processed and deleted (Rule [J3a]).

(4) Grammar fragment for /hasi/ transduction; q_ι final state

$$\begin{array}{lr}
[J1a] & \varepsilon() \rightarrow q_\varepsilon(\varepsilon()) \\
[J3a] & \mu(q_\varepsilon(t)) \rightarrow q_E(t) \\
[J4a] & \alpha(q_E(t_1), q_E(t_2)) \rightarrow q_\delta(L, H) \\
[J5a] & q_\delta(t_1, t_2) \rightarrow \\
 & q_\alpha(\alpha(t_1, t_2, \mu(t_1), \mu(t_2))) \\
[J6] & \iota(q_\alpha(t)) \rightarrow q_\iota(\iota(t, L))
\end{array}$$

When the α node is processed in Rule [J4a], the L and H tones are inserted. We could change the right-hand side of Rule [J4a] to $q_\alpha(\alpha(L, H, \mu(L), \mu(H)))$, skip Rule [J5a] entirely, and still generate the output of Rule [J5a]. But then the L tone that is daughter to the α node would have no specified relation to the L tone that is daughter to the left μ node; nor would the two H tones have a specified relation. Having instead the intermediate step of Rule [J4a] as written in (4) first inserts the L and H tones as lexical items and then carries them up the derivation to the next step as separate subtrees, without merging them at the α node.

Rule [J4a], which transitions the transducer to state q_δ, is our first example of a "multi" step—a step that carries multiple subtrees up the derivation rather than just one. The output of Rule [J4a] has a q_δ green circle that is not positioned as a mother node to a constituent because L and H have not been merged to build a constituent. The two daughter subtrees under q_δ in the input to Rule [J5a] also make that rule a "multi" rule. With L and H carried up as separate subtrees, Rule [J5a] builds μ constituents and an α constituent, associates the left daughter under q_δ (t_1) as both the leftmost daughter of the new α node and the daughter to the new leftmost μ node, and associates the right daughter under q_δ (t_2) as both the peninitial daughter of the new α node and the daughter to the new rightmost μ node. These multiple associations are represented with a multidominance structure, as discussed for Figure 2a. To end the derivation, Rule [J6] processes the ι node and adds an L sister to the right of the α subtree under ι.

"Multi" rules appear in the transduction for tonal assignment in *hasi* because the L and H tones each have multiple (two) dependencies in the course of the derivation. They each enter in Rule [J4a], when the α node is processed, but then they also enter relations with mother α and μ nodes in Rule [J5a]. While the output in Figure 2a derived by the *hasi* transduction shows multiple tonal associations, it

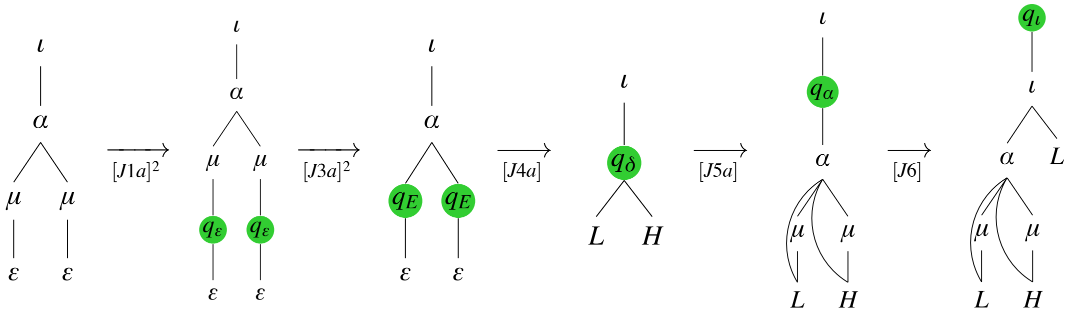


Table 2: Transduction for tonal association in unaccented /hasi/ using rules in (4)

is the multiple dependencies in the derivation steps that we are defining by including "multi" steps. The next transduction we show, which derives the output in Figure 2b, also has "multi" steps.

A transduction for a 2σ word with initial accent, e.g., *hási*, is given in Table 3—following rules already given in (4) and the additional rules in (5). As proposed in Pierrehumbert and Beckman (1988, p. 124-5), a T "tone" constituent is introduced. It keeps the two tones of the H*+L lexical accent as separate leaves so that the tones can dock onto separate TBUs. We assume here that the T node is deleted when the two tones dock onto separate TBUs, although alternative assumptions could be explored as well, see, e.g., Grice (1995).

(5) Grammar fragment for /hási/ transduction, not including rules in (4); q_ι final state

$$\begin{array}{lr}
[J1b] & H() \rightarrow q_H(H()) \\
[J1c] & L() \rightarrow q_L(L()) \\
[J2] & \mathrm{T}(q_H(t_1), q_L(t_2)) \rightarrow q_A(t_1, t_2) \\
[J3b] & \mu(q_A(t_1, t_2)) \rightarrow q_B((t_1, t_2)) \\
[J4b] & \alpha(q_B(t_1, t_2), q_E(t_3)) \rightarrow \\
 & q_\alpha(\alpha(L, \mu(t_1), \mu(t_2)))
\end{array}$$

Since accent in Japanese is lexical, unlike the Bengali pitch accent, the input tree to the transduction already has the first mora associated to a *T* subtree with *H* and *L* daughters, i.e., a lexical accent. The tonal leaves enter via Rules [J1a,b,c]. "Multi" Rule [J2] processes the *T* node and deletes it to expose the tonal daughters for tonal re-association, carrying the *H* and *L* leaves separately up the derivation. "Multi" Rule [J3b] processes the left μ node, deletes it, and continues to carry the *H* and *L* leaves up the derivation. Rule [J3a] processes the right μ node and deletes it. The *H* and *L* leaves have been carried up separately via the "multi" rules up to this point so that they can associate to separate moras in the next step. Rule [J4b] then shifts the *L* to the right branch to replace the placeholder ε, rebuilds the moras, and inserts an *L* accentual phrase tone as the leftmost sister to the μ's. Finally, Rule [J6] processes the ι node and inserts an *L* to the right of the α subtree as a daughter of ι, just like in the transduction for *hasi*.

The final output tree from Table 3 has no multiple tonal associations. Nevertheless, the transducer defined in (5) is an mbutt because "multi" steps arise from multiple dependencies in the derivation steps. Each tone of the lexical accent has two dependencies: (i) to the *T* node, where it enters as a daughter leaf, and (ii) to the μ node, where it re-associates as as daughter leaf. Note that the last "multi" step, Rule [J4b], can be easily modified to demonstrate how "multi" steps can accommodate non-peripheral temporal sequencing of edge tones. For example, the right-hand side could be changed to $q_\alpha(\alpha(\mu(t_1), L, \mu(t_2)))$ to insert the L_α between the two lexical accent tones.

4 Multiple dependencies in segmental associations

Multiple dependencies don't occur only with tonal association in prosodic trees, but also with prosodically-conditioned segmental processes. A segment enters as a leaf (location one in a prosodic tree) but then it cannot be merged into the prosodic tree until the prosodic constituent that conditions its realization is processed (location two). We illustrate this for the transduction of /r/-assimilation in the Bengali accentual phrase /t͡ʃador/, shown in Table 4, following (6). Recall from §1 that /r/ undergoes total assimilation to an immediately following coronal consonant within the same α. Therefore, the realization of any /r/ in Bengali can't be determined until α is processed. Moreover, any coronal consonant must also be carried up all the way to the α node, in case it may be immediately preceded by an /r/ within the same α.

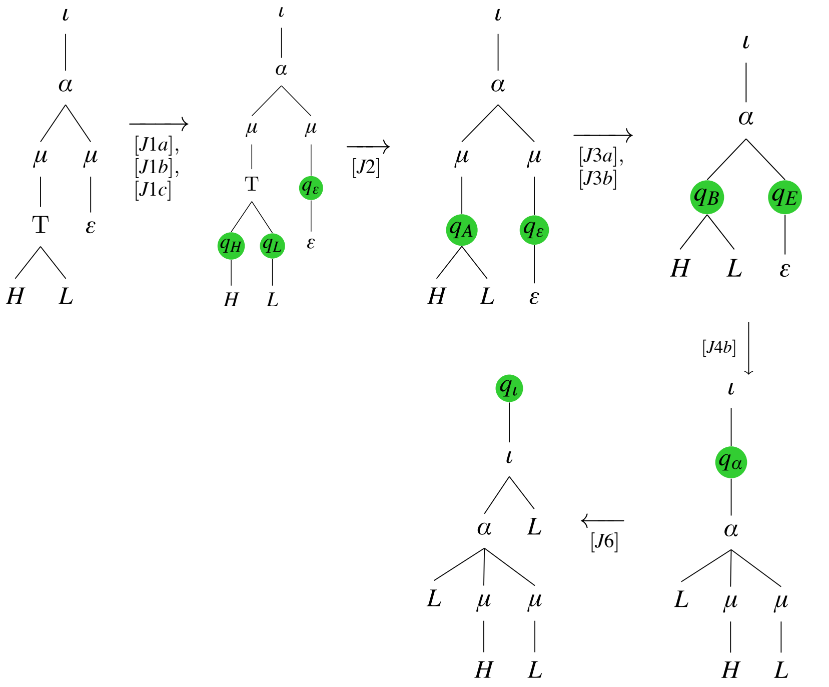


Table 3: Transduction for tonal association in accented /hási/ using rules in (5)

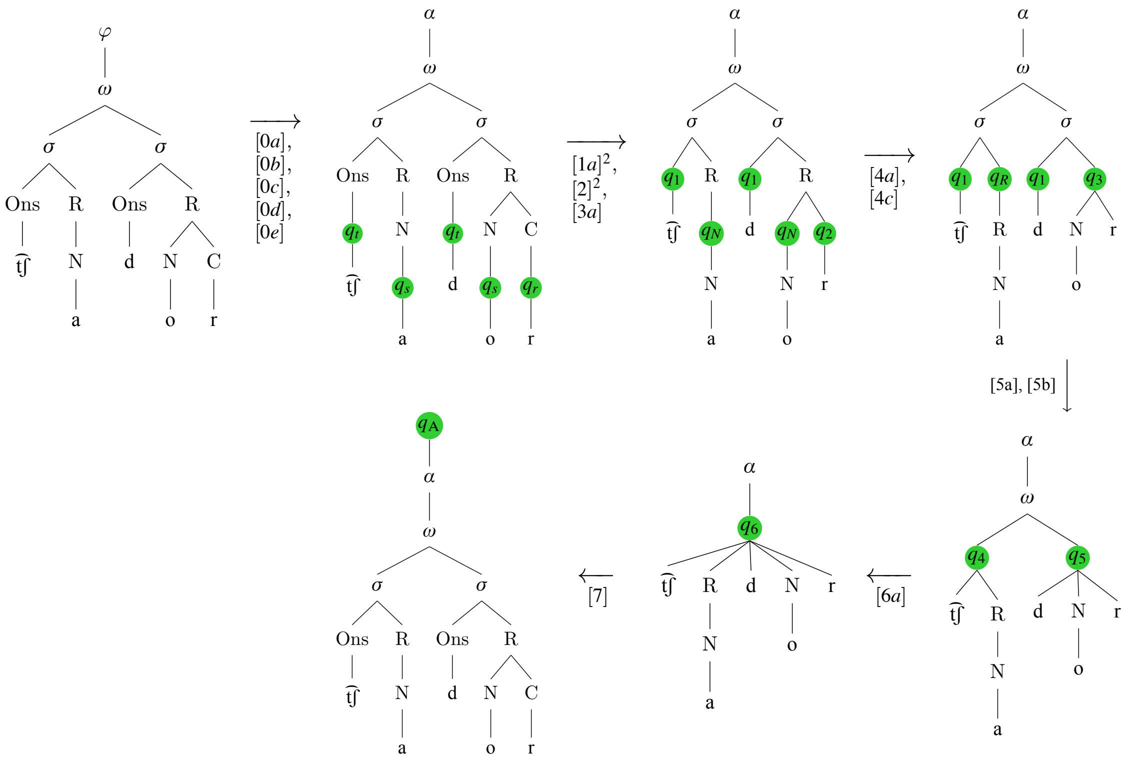


Table 4: Transduction for /r/-assimilation in /t͡ʃador/ using rules in (6)

(6) Grammar fragment for /t͡ʃador/; q_{A} final state

$$\begin{array}{lrlr}
[0a] & \text{t͡ʃ}() \rightarrow q_t(\text{t͡ʃ}()) & [0b] & \text{a}() \rightarrow q_s(\text{a}()) \\
[0c] & \text{d}() \rightarrow q_t(\text{d}()) & [0d] & \text{o}() \rightarrow q_s(\text{o}()) \\
[0e] & \text{r}() \rightarrow q_r(\text{r}()) & [1a] & \text{Ons}(q_t(t)) \rightarrow q_1(t) \\
[2] & \text{N}(q_s(t)) \rightarrow q_N(\text{N}(t)) & [3a] & \text{C}(q_r(t)) \rightarrow q_2(t)
\end{array}$$

$$\begin{array}{lr}
[4a] & \text{R}(q_N(t)) \rightarrow q_R(\text{R}(t)) \\
[4c] & \text{R}(q_N(t_1), q_2(t_2)) \rightarrow q_3(t_1, t_2) \\
[5a] & \sigma(q_1(t_1), q_R(t_2)) \rightarrow q_4(t_1, t_2) \\
[5b] & \sigma(q_1(t_1), q_3(t_2, t_3)) \rightarrow q_5(t_1, t_2, t_3) \\
[6a] & \omega(q_4(t_1, t_2), q_5(t_3, t_4, t_5)) \rightarrow q_6(t_1, t_2, t_3, t_4, t_5) \\
[7] & \alpha(q_6(t_1, t_2, t_3, t_4, t_5)) \rightarrow \\
 & q_{\text{A}}(\alpha(\omega(\sigma(O(t_1), t_2), \sigma(O(t_3), \text{R}(t_4, \text{C}(t_5))))))
\end{array}$$

The transduction in Table 4 begins by processing each segmental leaf via Rules [0a-e] to enter one of three states on each branch: q_t (coronals /t͡ʃ, d/), q_r (/r/), or q_s (vowels /a, o/). Then, each of the two branches with a nucleus (N) node with a daughter in state q_s can be processed to fix the realization of the segment and finish building the nucleus to enter state q_N (Rule [2]). However, any branch with a coronal or /r/ is processed to delete the onset (Ons) or coda (C) node and carry up the segment via Rules [$1a$, $3a$]. A unary rime (R) can then be built (Rule [4a]), since it doesn't have coronals or /r/. But the /r/ must continue to be passed up, so we delete its *R* mother node and then hold the /r/ together with a nucleus subtree without merging to enter state q_3 ("multi" Rule [4c]).

The coronals and /r/ (and already-built nuclei and rime) continue to be passed up as the σ nodes are processed and deleted ("multi" Rules [5a,b]) to reach states q_4 on the left branch (carrying up 2 subtrees) and q_5 on the right (carrying up 3 subtrees). Similarly, the ω node is then processed and deleted and the coronals and /r/ (and already-built nuclei and rime) are passed up again to enter q_6 with 5 subtree daughters ("multi" Rule [6a]). Finally, we are ready to process the α node and can stop passing up the coronals and /r/. Rule [7] (with Ons abbreviated as *O*) processes the α node and outputs an α tree with the remaining prosodic structure built in a single step, including no change to the final /r/, since the /r/ has no coronal sister to the immediate right under state q_6. Although /r/-assimilation does not apply when /t͡ʃador/ is in its own α, the transduction we just stepped through underscores that even if a coronal does not immediately follows an /r/ within the same α and even if an /r/ does not immediately precede a coronal within the same α, the dependency to the α node for these types of segments is always there. Appendix §C shows the tree transduction for /ɔmor t͡ʃador/ → [ɔmot͡ʃ t͡ʃador] when /ɔmor t͡ʃador/ is within the same α as in (1b,d).

5 Conclusion

We've shown that tree grammars defined via bottom-up tree transductions—standard and well-studied tools from formal language theory—provide a way to represent tonal association to higher-level nodes in prosodic trees. The peripherality of prosodic boundary tones follows without further stipulation (unlike Pierrehumbert and Beckman (1988)), since bottom-up tree transductions cannot change structures that have already been created. Extension to mbutts provides a mechanism to define non-peripheral boundary tones, which cannot be handled by Pierrehumbert and Beckman (1988). Since non-peripheral boundary tones such as Gussenhoven (2000)'s case in Roermond Dutch appear to be typologically rare, it seems desirable that non-peripheral boundary tones come in with the additional expressivity of "multi" steps in the grammar. More generally, mbutts can represent the pervasive multiple dependencies in prosodic structures including those arising in tonal association and from prosodically-conditioned segmental allophony. They offer a way to precisely state and probe proposals in phonological analyses of tone and intonation, at a time when the fundamental assumptions of AM theory are being revisited (Grice, 2021).

(M)butts are a good starting point since their computational properties are relatively well-understood, but the sample transductions shown here already reveal issues with using them for phonology. For one, the transductions exemplified here only define bounded structures, e.g., two-syllable prosodic words. We can introduce recursion into the grammar to build words and phrases of arbitrary length (Yu, 2021), but it remains to be seen how resulting self-embedded structures fit with phonological patterns. Another issue is that the restriction that (m)butts cannot modify already-built structure—while potentially desirable for making non-peripheral boundary tones possible but exceptional—results in mass deletion of structure in the derivation followed by rebuilding this structure in a single step. Much more work is needed to refine, restrict, and adapt (m)butts to capture and identify generalizations about prosodic structure.

References

Amalia Arvaniti and Janet Fletcher. 2020. The Autosegmental-Metrical theory of intonational phonology. In Carlos Gussenhoven and Aoju Chen, editors, *The Oxford Handbook of Language Prosody*, chapter 6, pages 78–95. Oxford University Press.

Brenda Baker. 1978. Tree transducers and tree languages. *Information and control*, 37:241–266.

Jane Chandlee and Adam Jardine. 2021. Input and output locality and representation. *Glossa*, 6(1):43.

Alexander Clark. 2014. An introduction to multiple context-free grammars for linguists. Available at: `http://www.cs.rhul.ac.uk/home/alexc/lot2012/mcfgsforlinguists.pdf`.

H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. 2007. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`. Release October, 12th 2007.

Hossep Dolatian, Aniello De Santo, and Thomas Graf. 2021. Recursive prosody is not finite-state. In *Proceedings of SIGMORPHON 2021*.

Joost Engelfriet, Eric Lilin, and Andreas Maletti. 2009. Extended multi bottom–up tree transducers: Composition and decomposition. *Acta Informatica*, 46(8):561–590.

Zoltán Fülöp, Armin Kühnemann, and Heiko Vogler. 2004. A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead. *Information Processing Letters*, 91:57–67.

Hans-Martin Gärtner. 2002. *Generalized Transformations and Beyond*. Akademie Verlag, Berlin.

John Anton Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

Thomas Graf. 2012. Movement-generalized Minimalist Grammars. In *LACL 2012*, volume LNCS 7351, pages 58–73.

Martine Grice. 1995. Leading tones and downstep in English. *Phonology*, 12(2):183–233.

Martine Grice. 2021. Commentary: The autosegmental-metrical model of intonational phonology. In Stefanie Shattuck-Hufnagel and Jonathan A. Barnes, editors, *Prosodic Theory and Practice*. MIT Press, Cambridge, MA.

Carlos Gussenhoven. 2000. The boundary tones are coming: on the nonperipheral realization of boundary tones. In Michael B. Broe and Janet B. Pierrehumbert, editors, *Papers in Laboratory Phonology V: Acquisition and the lexicon*, pages 132–151. Cambridge University Press, Cambridge, UK.

Carlos Gussenhoven. 2004. *The phonology of tone and intonation*. Cambridge University Press, Cambridge, UK.

Carlos Gussenhoven. 2014. On the privileged status of boundary tones: evidence from Japanes, French, and Cantonese English. In Eugene Buckley, Thera Crane, and Jeff Good, editors, *Revealing structure*, pages 1–13. CSLI Publications.

Carlos Gussenhoven. 2018. Prosodic typology meets phonological representations. In Larry M. Hyman and Frans Plank, editors, *Phonological typology*, pages 389–418. Walter de Gruyter GmbH, Berlin/Boston.

Carlos Gussenhoven. To appear. Just how metrical is the Autosegmental-Metrical model? Evidence from pitch accents in Nubi, Persian, and English. In Haruo Kubozono, Junko Ito, and Armin Mester, editors, *Prosody and Prosodic Interfaces*. Oxford University Press.

Bruce Hayes and Aditi Lahiri. 1991. Bengali intonational phonology. *Natural Language & Linguistic Theory*, 9:47–96.

Sameer ud Dowla Khan. 2008. *Intonational phonology and focus prosody of Bengali*. Ph.D. thesis, University of California Los Angeles.

Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In J. Rogers and S. Kepser, editors, *Model Theoretic Syntax at 10*, pages 71–80.

D. Robert Ladd. 1996. *Intonational phonology*. Cambridge University Press, Cambridge.

Eric Lilin. 1978. *Une generalisation des transducteurs d'etats finis d'arbres: les S-transducteurs*. Thése 3éme cycle, Université de Lille.

Andreas Maletti. 2008. Compositions of extended top-down tree transducers. *Information and computation*, 206:1187–1196.

Janet Pierrehumbert. 2011. Representation in phonology. Presented at 50 Years of Linguistics at MIT.

Janet Pierrehumbert and Mary Beckman. 1988. *Japanese Tone Structure*. The MIT Press.

J.B. Pierrehumbert. 1980. *The phonology and phonetics of English intonation*. Ph.D. thesis, Massachusetts Institute of Technology.

Pilar Prieto, Mariapaola D'Imperio, and Barbara Gili Fivela. 2005. Pitch accent alignment in Romance: primary and secondary associations with metrical structure. *Language and Speech*, 48:359–396.

William C. Rounds. 1970. Mappings and grammars on trees. *Mathematical Systems Theory*, 4(3):257–287.

Kristine M. Yu. 2021. Computational perspectives on phonological constituency and recursion. *Catalan Jouurnal of Linguistics*, 20:77–114.

A Warm-up tree transduction for only accentual phrase tone insertion

This "warm-up" bottom-up tree transduction inserts only an accentual phrase tone in a single word-accentual phrase in Bengali, while ignoring stress assignment and pitch accent assignment. We include it to show an example of a bottom-up tree transduction that only inserts material in building the output tree. In contrast, the transduction given in Table 1 and (3) includes the step of Rule [B2] which deletes the σ node.

(7) Grammar fragment for tree transduction of single-ω accentual phrase without pitch accent; q_α final state

$$\begin{array}{lr}
[B1] & \varepsilon() \rightarrow q_\varepsilon(\varepsilon()) \\
[B2a] & \sigma(q_\varepsilon(t) \rightarrow q_\sigma(\sigma(t)) \\
[B3a] & \omega(q_\sigma(t_1), q_\sigma(t_2)) \rightarrow q_\omega(\omega(t_1, t_2)) \\
[B4] & \alpha(q_\omega(t)) \rightarrow q_\alpha(\alpha(t, H))
\end{array}$$

Rules [B1, B4] are already discussed in §2 and we do not repeat discussion of them here. Rule [B2a] states that if the transducer is at a unary σ node with its single daughter (the variable t) in state q_ε, then the transducer can process the σ node and enter state q_σ, leaving its daughter (the variable t, and in this case, ε) unchanged. The second step in Table 5 shows the transducer applying Rule [B2a] for both the left and right branches. Rule [B3a] is a merge rule that states that if the transducer is at a binary-branching ω node with its left daughter (t_1) in state q_σ and its right daughter (t_2) also in state q_σ, then the transducer can process the ω node to enter q_ω and output back the ω subtree without any change to daughters t_1, t_2.

B Tree transduction for tonal association in two-ω accentual phrase

Table 1 and (3) in §2 made the simplification that the pitch accent insertion rule in Bengali is only ω-based, i.e., a pitch accent is assigned to the stressed syllable in each ω. But in fact, pitch accent assignment is both ω- and accentual-phrase based, i.e., a pitch accent is only assigned to the stressed syllable of an α-initial ω. The rules in (8) and the tree transduction in Table 6 show one way this can be done. The leftmost ω that is pitch-accented is built with steps in Table 1, but the rightmost ω, which is unaccented, uses another rule, Rule [B3b]. Note that since Rules [B3] and [B3b] share the same left hand side, there is non-determinism in the grammar and either of the rules could apply when an ω node is processed in the second step of the transduction. However, Rule [B4b] restricts well-formed two-ω accentual phrases to being initially accented.

(8) Grammar fragment for tree transduction of two-ω accentual phrase, repeating rules already in (3); q_α final state

$$\begin{array}{lr}
[B1] & \varepsilon() \rightarrow q_\varepsilon(\varepsilon()) \\
[B2] & \sigma(q_\varepsilon(t)) \rightarrow q_\zeta(t) \\
[B3] & \omega(q_\zeta(t_1), q_\zeta(t_2)) \rightarrow q_\omega(\omega(\mathrm{str}(\sigma(L)), \sigma(t_2))) \\
[B3b] & \omega(q_\zeta(t_1), q_\zeta(t_2)) \rightarrow q_\Omega(\omega(\mathrm{str}(\sigma(t_1)), \sigma(t_2))) \\
[B4b] & \alpha(q_\omega(t_1), q_\Omega(t_2)) \rightarrow q_\alpha(\alpha(t_1, t_2, H))
\end{array}$$

C Tree transduction for /r/-assimilation in /ɔmor t͡ʃador/ → [ɔmot͡ʃ t͡ʃador]

Some rules below are repeated from the rules for the /t͡ʃador/ transduction in (6).

First, we show the transduction for a single-word accentual phrase for /ɔmor/ in Table 7, using the rules in (9). Like /t͡ʃador/, /ɔmor/ has a /r/ that needs to be passed up to the α node.

(9) Grammar fragment for /ɔmor/ transduction; q_α final state;

$$\begin{array}{lr}
[0d] & \mathrm{o}() \rightarrow q_s(\mathrm{o}()) \\
[0e] & \mathrm{r}() \rightarrow q_r(\mathrm{r}()) \\
[0h] & \text{ɔ}() \rightarrow q_s(\text{ɔ}()) \\
[0i] & \mathrm{m}() \rightarrow q_s(\mathrm{m}()) \\
[1] & \mathrm{Ons}(q_s(t)) \rightarrow q_O(\mathrm{Ons}(t)) \\
[2] & \mathrm{N}(q_s(t)) \rightarrow q_N(\mathrm{N}(t)) \\
[3a] & \mathrm{C}(q_r(t)) \rightarrow q_2(t) \\
[4a] & \mathrm{R}(q_N(t)) \rightarrow q_R(\mathrm{R}(t)) \\
[4c] & \mathrm{R}(q_N(t_1), q_2(t_2)) \rightarrow q_3(t_1, t_2) \\
[5c] & \sigma(q_O(t_1), q_3(t_2, t_3)) \rightarrow q_7(t_1, t_2, t_3) \\
[5d] & \sigma(q_R(t)) \rightarrow q_\sigma(\sigma(t)) \\
[6c] & \omega(q_\sigma(t_1), q_7(t_2, t_3, t_4)) \rightarrow q_9(t_1, t_2, t_3, t_4) \\
[7b] & \alpha(q_9(t_1, t_2, t_3, t_4)) \rightarrow \\
& q_\alpha(\alpha(\omega(\mathrm{str}(t_1), \sigma(\mathrm{Ons}(t_2), \mathrm{R}(t_3, \mathrm{C}(t_4)))))))
\end{array}$$

Putting together the tranductions of /ɔmor/ and /t͡ʃador/ in Tables 7 and 4 up through the penultimate steps, we can define the transduction of /r/-assimilation in the single accentual phrase /ɔmor t͡ʃador/ in Table 8 with the additional rule in (10).

(10) Grammar fragment for transduction of /r/-assimilation in /ɔmor t͡ʃador/; q_α final state

$$\begin{array}{r}
[7c]\ (q_9(t_1, t_2, t_3, t_4), q_6(t_5, t_6, t_7, t_8, t_9)) \rightarrow \\
q_\alpha(\alpha(\omega(t_1, \sigma(t_2, \mathrm{R}(t_3, \mathrm{C}(t_5)))), \\
\omega(\sigma(\mathrm{Ons}(t_5), t_6), \sigma(\mathrm{Ons}(t_7), \mathrm{R}(t_8, \mathrm{C}(t_9))))))
\end{array}$$

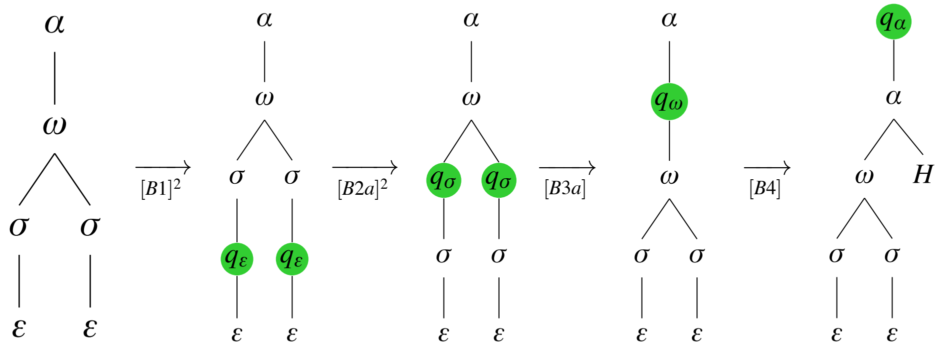


Table 5: Transduction of accentual phrase tone insertion in single-word accentual phrase using rules in (7)

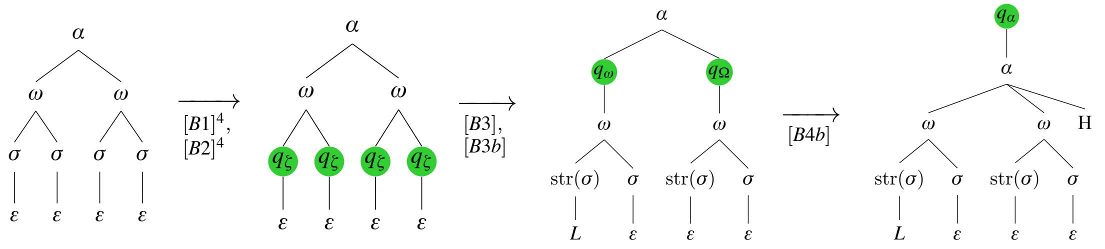


Table 6: Transduction of tone insertion and stress assignment in two-word accentual phrase using rules in (8)

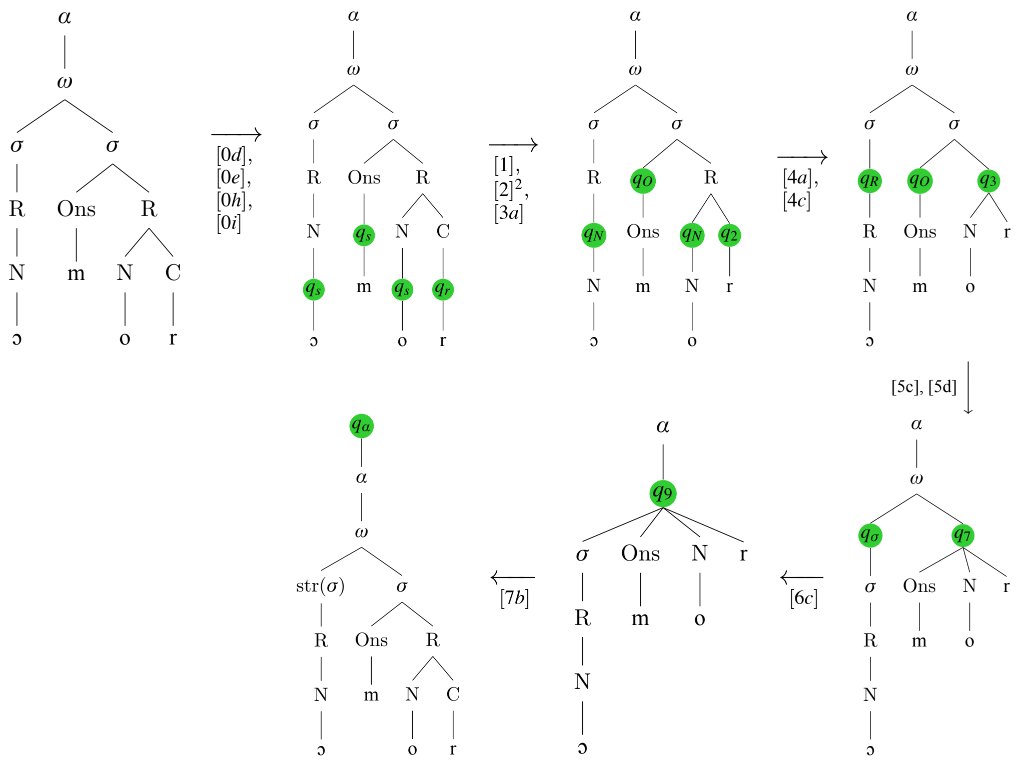


Table 7: Transduction for /r/-assimilation in /ɔmor/ using rules in (9)

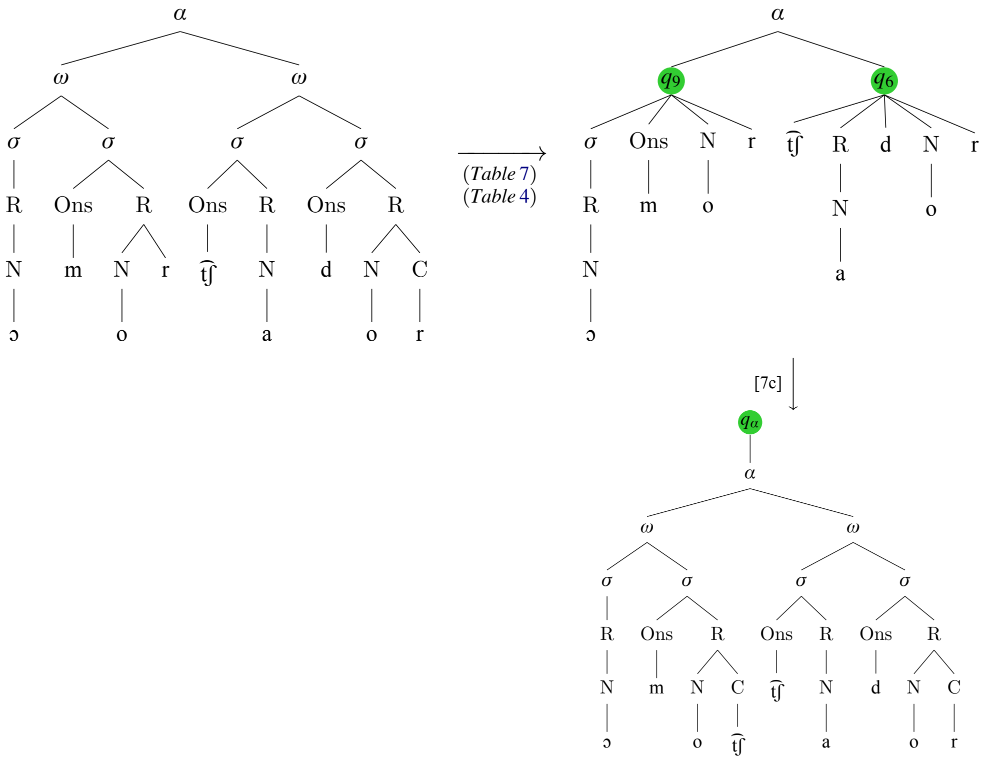


Table 8: Transduction for /r/-assimilation in /ɔmor t͡ʃador/ using rules in (10)

Typological Implications of Tier-Based Strictly Local Movement

Thomas Graf
Stony Brook University
Department of Linguistics
100 Nicolls Road, Stony Brook, NY 11794, USA
mail@thomasgraf.net

Abstract

Earlier work has shown that movement, which forms the backbone of Minimalist syntax, belongs in the subregular class of TSL-2 dependencies over trees. The central idea is that movement, albeit unbounded, boils down to local mother-daughter dependencies on a specific substructure called a tree tier. This reveals interesting parallels between syntax and phonology, but it also looks very different from the standard view of movement. One may wonder, then, whether the TSL-2 characterization is linguistically natural. I argue that this is indeed the case because TSL-2 furnishes a unified analysis of a variety of phenomena: multiple wh-movement, expletive constructions, the *that*-trace effect and the anti-*that*-trace effect, islands, and wh-agreement. In addition, TSL-2 explains the absence of many logically feasible yet unattested phenomena. Far from a mere mathematical curiosity, TSL-2 is a conceptually pleasing and empirically fertile characterization of movement.

1 Introduction

A number of recent works (Graf 2018; Graf and De Santo 2019; Vu et al. 2019; Shafiei and Graf 2020; Graf and Kostyszyn 2021, a.o.) have investigated the complexity of syntax from a subregular perspective. One of the central findings is that movement as formalized in Minimalist grammars (Stabler, 1997, 2011) is *tier-based strictly 2-local* (TSL-2). This means that one can determine whether a movement step in a syntactic derivation is well-formed by I) constructing a tree tier that only contains material relevant to this kind of movement, and ii) checking mother-daughter configurations over this tree tier. But the specific system for movement is just one among many options that could be expressed in TSL-2. This raises questions about the empirical status of those other options, and whether they ever occur in language. In this paper, I argue that TSL-2 provides a broad typology of movement in the sense that every architectural option it provides is actually used with some movement-related phenomenon: multiple wh-movement, expletive constructions, the *that*-trace effect and the anti-*that*-trace effect, islands, and wh-agreement in Irish.

All of these phenomena, many of which are puzzling under the standard conception of Minimalist movement, fall out naturally from the TSL-perspective. The central argument is that if a cognitive system must be TSL-2 to handle movement, then we should expect to see these TSL-2 resources be used in a variety of ways. For instance, if the complexity of a system with movement and island constraints is not higher than that of the movement system without island constraints, additional explanations would be needed if no language ever exhibited island effects. Island constraints would inevitably be part of a linguistic ecosystem of free variation that is limited only by the available cognitive resources. Free variation limited to TSL-2 thus carves out a space within which we find some of the most surprising movement phenomena.

The paper is primarily a progression of case studies. The necessary background of TSL-2 movement is covered in §2. I then summarize earlier arguments by Graf and Kostyszyn (2021) that multiple wh-movement and expletive constructions are also TSL-2 (§3.1) before I turn to a new TSL-2 analysis of the *that*-trace effect (§3.2) that, among other things, hinges on the ability to put non-movers on movement tiers. I subsequently generalize this technique to also handle island effects (§4.1) and even wh-agreement (§4.2). All of this establishes that the space of TSL-2 dependencies includes a large variety of movement phenomena. But as discussed in §5, there is still overgeneration within this space, and some movement phenomena do seem to fall outside TSL-2. This should not prove to be an insur-

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 184-193.
Held on-line February 7-9, 2022

mountable challenge, though, and I propose several ways this could be addressed in future research.

2 Tier-based strictly local movement

The TSL-view of syntax builds on Minimalist grammars (MGs; Stabler, 1997, 2011), which are a formalization of Minimalist syntax. Every lexical item (LI) is annotated with features that determine its syntactic behavior. At the very least, each LI has some *category feature* $\texttt{F}^-$, for instance in the noun $\texttt{party} :: \texttt{N}^-$. An LI may also have a string of *selector features* $\texttt{F}_1^+ \cdots \texttt{F}_m^+$ that determine which arguments it takes. An example would be the ditransitive verb $\texttt{introduce} :: \texttt{P}^+\texttt{D}^+\texttt{D}^+\texttt{V}^-$ as in *John introduced Mary to Sue*. In addition, an LI may carry *licensor features* $\texttt{f}_1^+ \cdots \texttt{f}_n^+$, which provide a landing site for movement. In this paper, no LI will ever have more than one licensor feature — consider for instance the empty topicalization head $\varepsilon :: \texttt{T}^+\texttt{top}^+\texttt{C}^-$, with a single licensor feature $\texttt{top}^+$ that attracts a topicalized phrase. Finally, an LI may carry a set $\{\texttt{f}_1^-, \ldots, \texttt{f}_n^-\}$ of unordered *licensee features* (standard MGs assume that licensee features are also linearly ordered, but this is incompatible with the TSL-view of syntax; as is already implicit in Graf et al. 2016, the use of unordered licensee features does not alter the weak or strong generative capacity of MGs). Each licensee feature $\texttt{f}^-$ on LI l indicates that the phrase headed by l moves to the closest landing site provided by an LI with $\texttt{f}^+$. Each LI thus has a feature annotation of the form $\gamma\texttt{F}^-\delta$, where γ is a (possibly empty) string of selector and licensor features, $\texttt{F}^-$ is some category feature, and δ is either the empty string or a set of licensee features.

The syntactic derivations driven by those features can be succinctly represented in the form of a dependency tree as shown in Fig. 1. Movement in this formalism is *tier-based strictly 2-local* (TSL-2). A full definition of TSL-2 over trees is given in Graf and Kostyszyn (2021), but an intuitive discussion suffices for the purposes of this paper. I will first discuss TSL-2 over strings and then explain how this idea is generalized to trees.

TSL-2 over strings was first defined in Heinz et al. (2011) and is a generalization of the class *strictly 2-local* (SL-2). A stringset L is SL-2 iff there is a finite (and possibly empty) set G of forbidden bigrams such that L contains all strings s, and only those, such that $\rtimes s \ltimes$ does not contain any of G's forbidden bigrams. Here $\rtimes$ and $\ltimes$ are distinguished symbols that mark the beginning and end of the string, respectively. A well-known SL-2 stringset is $(ab)^+$, which contains ab, $abab$, and so on. It is SL-2 because it can be described by 5 forbidden bigrams (assuming that the alphabet is already limited to just a and b, otherwise additional bigrams are needed):

(1) a. $\rtimes\ltimes$: the string must contain at least one symbol

b. $\rtimes b$: the string must not start with b

c. aa: a must not be followed by a

d. bb: b must not be followed by b

e. $a\ltimes$: the string must not end with a

As another example, suppose that we only consider strings over the symbol a. Then all of the following stringsets are SL-2:

(2) a. the set of all strings over a ($G := \emptyset$)

b. the set of all strings with no a (G contains at least $\rtimes a$ or $a\ltimes$)

c. the set of all strings with at least one a ($G := \{\rtimes\ltimes\}$)

d. the set of all strings with at most one a ($G := \{aa\}$)

e. the set of all strings with exactly one a ($G := \{\rtimes\ltimes, aa\}$)

Intuitively, SL-2 models string dependencies that can be expressed as a finite number of constraints where one symbol restricts what other symbols may immediately occur to its right. TSL-2 over strings enriches SL-2 with a tier projection mechanism to allow for limited types of long-distance dependencies. Formally, tier projection is expressed as a function E_T that takes a string s as its input and deletes all symbols in s that do not belong to T. For example, $E_{\{a,b\}}$ would map $caccbac$ to aba. A stringset L is TSL-2 iff there is some finite tier alphabet T such that the image of L under E_T is SL-2. For instance, the set of strings over a and b that contain exactly one a is not SL-2, but it is TSL-2: we set $T := \{a\}$ and $G := \{\rtimes\ltimes, aa\}$. Then the well-formed $babbb$ has the well-formed tier a (or $\rtimes a \ltimes$ with explicit edge markers), whereas the illicit $babab$ has the ill-formed tier aa. TSL-2 thus captures the notion that long-distance dependencies are still local when irrelevant material is ignored.

TSL-2 over trees follows a very similar system of combining an SL-mechanism with a tier projection. Given a finite set T of tier symbols, one

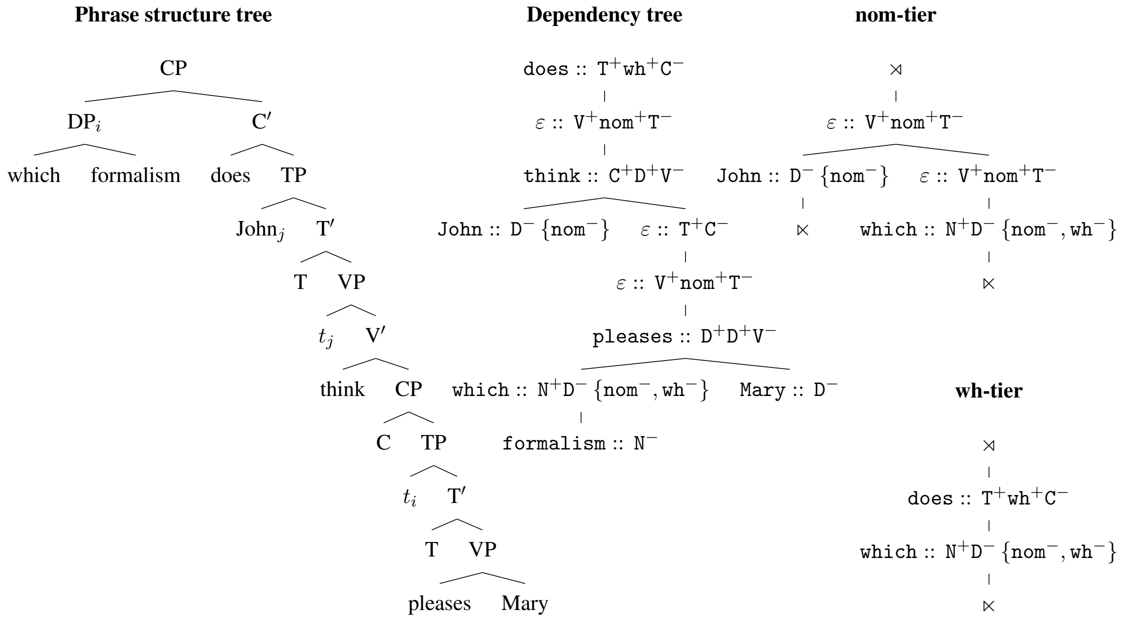

Phrase structure tree
CP
DP_i
C′
which
formalism
does
TP
John_j
T′
T
VP
t_j
V′
think
CP
C
TP
t_i
T′
T
VP
pleases
Mary
Dependency tree
$\mathtt{does} :: \mathtt{T}^+\mathtt{wh}^+\mathtt{C}^-$
$\varepsilon :: \mathtt{V}^+\mathtt{nom}^+\mathtt{T}^-$
$\mathtt{think} :: \mathtt{C}^+\mathtt{D}^+\mathtt{V}^-$
$\mathtt{John} :: \mathtt{D}^- \{\mathtt{nom}^-\}$
$\varepsilon :: \mathtt{T}^+\mathtt{C}^-$
$\varepsilon :: \mathtt{V}^+\mathtt{nom}^+\mathtt{T}^-$
$\mathtt{pleases} :: \mathtt{D}^+\mathtt{D}^+\mathtt{V}^-$
$\mathtt{which} :: \mathtt{N}^+\mathtt{D}^- \{\mathtt{nom}^-, \mathtt{wh}^-\}$
$\mathtt{Mary} :: \mathtt{D}^-$
$\mathtt{formalism} :: \mathtt{N}^-$
nom-tier
$\rtimes$
$\varepsilon :: \mathtt{V}^+\mathtt{nom}^+\mathtt{T}^-$
$\mathtt{John} :: \mathtt{D}^- \{\mathtt{nom}^-\}$
$\varepsilon :: \mathtt{V}^+\mathtt{nom}^+\mathtt{T}^-$
$\ltimes$
$\mathtt{which} :: \mathtt{N}^+\mathtt{D}^- \{\mathtt{nom}^-, \mathtt{wh}^-\}$
$\ltimes$
wh-tier
$\rtimes$
$\mathtt{does} :: \mathtt{T}^+\mathtt{wh}^+\mathtt{C}^-$
$\mathtt{which} :: \mathtt{N}^+\mathtt{D}^- \{\mathtt{nom}^-, \mathtt{wh}^-\}$
$\ltimes$

Figure 1: Phrase structure tree (left) with corresponding annotated derivation tree (middle) and two well-formed movement tiers (right), each one containing exactly one LI with $\mathtt{f}^-$ among the daughters of each LI with $\mathtt{f}^+$; note that intermediate movement of *which formalism* to Spec,CP of the embedded clause is not encoded via features; $\rtimes$ and $\ltimes$ on tiers will be omitted for the rest of the paper

removes from the tree all nodes whose labels do not belong to T, while preserving dominance relations between the remaining nodes. On the tree tier, each mother may restrict the shape of its daughters, similar to how in SL-2 over strings a symbol may restrict the shape of the symbol immediately following it. Formally, each tier symbol σ in T is associated with a stringset L_σ, and if a node on the tier is labeled σ, then its daughters on the tier must form a string that belongs to L_σ.

MG movement fits into this general system as follows: For each movement type `f` (`nom`, `wh`, and so on) one removes all nodes from the dependency tree that do not carry at least one of $\mathtt{f}^+$ and $\mathtt{f}^-$. The result is the tree tier for `f` (cf. Fig. 1). In analogy to the string case, the tier also has a distinguished root $\rtimes$, and each leaf is made a mother of $\ltimes$. On the tier, each tier symbol σ is associated with a particular daughter stringset L_σ that is TSL-2: If n is a node on an `f`-tier and n has a label that includes $\mathtt{f}^+$, then the daughter string of n must contain exactly one node whose label includes $\mathtt{f}^-$. If n is labeled $\rtimes$, instead, then its daughter string must not contain any $\mathtt{f}^-$. This results in a system where both of the following hold for each `f`-tier: I) every $\mathtt{f}^+$-node has exactly one $\mathtt{f}^-$-daughter, and II) every $\mathtt{f}^-$-daughter has a $\mathtt{f}^+$-mother. That is exactly how movement behaves in MGs, making it "doubly TSL-2": it is TSL-2 over trees, and on each movement tier it holds for every node that its set of well-formed daughter strings is TSL-2.

But this TSL-2 view of movement allows for several alternatives of the same formal complexity. As previously illustrated in (2), TSL-2 can perform limited counting, distinguishing between 0, "at least 1", "at most 1", and "exactly 1". In standard MGs, the daughter string of an LI with $\mathtt{f}^+$ must contain exactly one LI with $\mathtt{f}^-$, but from the view of TSL-2 one could just as well require at least one $\mathtt{f}^-$, at most one, or none at all. In addition, $\mathtt{f}^+$ and $\mathtt{f}^-$ are meaningless symbols from the perspective of TSL-2, and thus there is no inherent reason why only LIs with those features should be present on a tier. And once these LIs appear on a tier, they could behave like LIs with $\mathtt{f}^+$ in that they put constraints on their daughters, or like LIs with $\mathtt{f}^-$ in that they can satisfy those constraints. The rest of this paper explores this typology of grammatical options carved out by TSL-2. I will show how varying these TSL-2 parameters yields various phenomena related to movement, which suggests that the TSL-2 characterization of movement isn't just a mathematical coincidence but touches on fundamental properties of movement.

3 Varying the number of dependents

I first consider the configurations that arise if one changes how many $\mathtt{f}^-$ have to occur in the daughter string. I argue that this yields multiple wh-movement, optional movement, and the *that*-trace effect in English (including exceptions brought about by adjuncts). The first two were already discussed in Graf and Kostyszyn (2021), so I will sketch them only briefly.

3.1 Multiple wh-movement and optional movement

Multiple wh-movement refers to the phenomenon where multiple wh-phrases move to the left edge of the clause

(3) *Multiple wh-movement in Bulgarian* (Bošković, 2002, p.353)

[Ko$_i$ koga$_j$ [t_i voli t_j]]?
who whom loves

'Who loves whom?'

In terms of TSL, this can be analyzed as a relaxation of movement where the matrix C-head ε :: $\mathtt{T^+wh^+C^-}$ still carries only one instance of $\mathtt{wh}^+$, but its string of daughters on the wh-tier may contain any number of wh-movers with $\mathtt{f}^-$, as long as it contains at least one (see Fig. 2). Since this is a weakening of the standard constraint ("exactly one" is equivalent "at least one and at most one"), the TSL-2 account of movement tells us that multiple wh-movement is unremarkable in the sense that a system that can require the presence of exactly one mover can also enforce the presence of at least one.

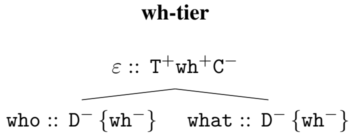


Figure 2: Example of wh-tier with multiple wh-movement

If, on the other hand, the requirement is loosened to "at most one", one gets a landing site that does not need a mover but can accommodate one if necessary — in other words, optional movement. Graf and Kostyszyn (2021) argue that this provides an alternative explanation of expletive constructions.

(4) a. A man is in the garden.
b. There is a man in the garden.

In (4a), the T-head ε :: $\mathtt{V^+nom^+T^-}$ has a matching nom-tier daughter a :: $\mathtt{N^+D^-}$ $\{\mathtt{nom^-}\}$, and movement takes place as usual. If *a* loses its licensee feature, one gets (4b) instead, where the T-head has no suitable daughter on the nom-tier, causing the unmatched $\mathtt{nom}^+$ to be spelled out as the expletive *there*. Again a well-known movement phenomenon has a natural place in the TSL-2 formalism.

3.2 The *that*-trace effect

The *that*-trace effect refers to the phenomenon that even though English allows for long-distance extraction from an embedded clause, subjects may not be extracted if the complementizer is *that*. Curiously, this effect disappears if *that* is followed by an adverb (cf. Browning, 1996, p.238).

(5) a. Who$_i$ do you think (that) John should have met t_i?
b. Who$_i$ do you think (*that) t_i should have t_i met John?
c. Who$_i$ do you think (that) under normal circumstances t_i should have t_i met John?

This can be analyzed in various ways, e.g. as a string constraint against *that* t. But TSL can accommodate this phenomenon without additional machinery.

Let us ignore the effect of adverbs for now. Suppose that we construct a wh-tier in the usual manner to verify that there is a match between wh-mover and wh-landing site. But in addition, we also construct another tier whose job it is to further restrict the behavior of subjects, thus giving rise to the *that*-trace effect. This *that-trace tier* (TTT) contains all of the following: I) every LI with $\mathtt{wh}^+$, II) every LI with both $\mathtt{nom}^-$ and $\mathtt{wh}^-$, and III) every C-head, including that :: $\mathtt{T^+C^-}$. Only one constraint is active on TTT, namely that the complementizer *that* must not have any LI among its daughters that carries $\mathtt{nom}^-$.

As shown in Fig. 3, this system correctly rules out the illicit *Who do you think that met John* while still allowing for well-formed counterparts that do not involve extraction of a subject wh-phrase. This account works thanks to the interaction of three factors. First, we can correctly pick out wh-subjects by their features $\mathtt{nom}^-$ and $\mathtt{wh}^-$, so that only subjects (but not objects) are projected onto TTT. Second, by also projecting $\mathtt{wh}^+$ nodes we introduce a safety buffer on TTT that pushes wh-subjects out of the

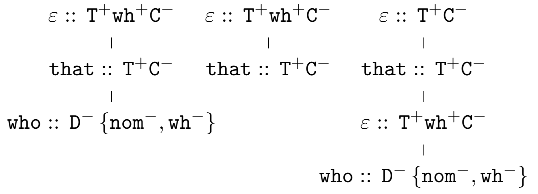

ε :: $\mathtt{T^+wh^+C^-}$
that :: $\mathtt{T^+C^-}$
who :: $\mathtt{D^-}$ {$\mathtt{nom^-, wh^-}$}
ε :: $\mathtt{T^+wh^+C^-}$
that :: $\mathtt{T^+C^-}$
ε :: $\mathtt{T^+C^-}$
that :: $\mathtt{T^+C^-}$
ε :: $\mathtt{T^+wh^+C^-}$
who :: $\mathtt{D^-}$ {$\mathtt{nom^-, wh^-}$}

Figure 3: Ill-formed TTT for illicit *Who do you think that met John* (left) and well-formed TTTs for licit *Who do you think that John met* and *I know that Mary wondered who met Bill* (middle and right)

daughter string of *that* if their wh-movement does not actually cross the complementizer. Finally, by projecting every C-head, including empty ones, we allow subject-wh phrases to cross *that* as long as their immediately containing clause has a different complementizer. This allows for well-formed examples such as the one below.

(6) Who$_i$ do you think that Mary said that John believes [C t_i met Bill]?

As the reader might have already noticed, the ameliorating effect of adverbs could be captured by projecting them onto TTT so that they separate subject wh-phrases from *that*. The big puzzle is how one wants to represent adverbs, which are adjuncts, in dependency trees. While the MG literature furnishes many different implementations of adjunction (see Frey and Gärtner 2002, Graf 2014, and Hunter 2015, a.o.), the easiest option in this case is *category-preserving selection*. That is to say, adjunction of some YP to XP is expressed as selection by an empty head ε :: $\mathtt{X^+Y^+X^-}$ that projects another XP. This is illustrated in Fig. 4. Since no other empty heads ever seem to display the particular feature pattern $\mathtt{T^+X^+T^-}$, the projection for TTT can correctly single out these TP-adjunction heads. But projecting TP-adjunction heads onto TTT can push the wh-subject out of the daughter string of *that*, and in this case TTT will be well-formed.

The reader may object that this is a highly stipulative proposal, but quite the opposite is the case. No stipulations are involved at all. TSL-2 carves out a space of options, and what this section shows is that both the *that*-trace effect and its exceptions are already part of this space. Individual points within the space may look highly peculiar, but the whole space itself is very natural.

Overall, then, the existence of the *that*-trace effect is unsurprising in the sense that it requires no additional machinery, assumptions, or stipulations

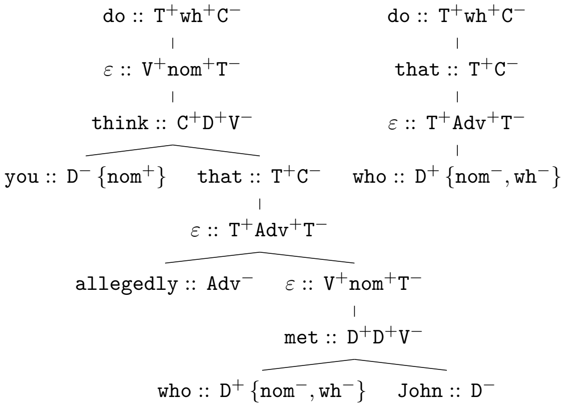

do :: $\mathtt{T^+wh^+C^-}$
ε :: $\mathtt{V^+nom^+T^-}$
think :: $\mathtt{C^+D^+V^-}$
you :: $\mathtt{D^-}$ {$\mathtt{nom^+}$}
that :: $\mathtt{T^+C^-}$
ε :: $\mathtt{T^+Adv^+T^-}$
allegedly :: $\mathtt{Adv^-}$
ε :: $\mathtt{V^+nom^+T^-}$
met :: $\mathtt{D^+D^+V^-}$
who :: $\mathtt{D^+}$ {$\mathtt{nom^-, wh^-}$}
John :: $\mathtt{D^-}$
do :: $\mathtt{T^+wh^+C^-}$
that :: $\mathtt{T^+C^-}$
ε :: $\mathtt{T^+Adv^+T^-}$
who :: $\mathtt{D^+}$ {$\mathtt{nom^-, wh^-}$}

Figure 4: In the dependency tree for *who$_i$ do you think that allegedly t_i met John* (left), projecting $\sim$ creates a buffer between *that* and *who* (right).

beyond what is already furnished by TSL-2. It admittedly requires a very particular choice of tier projections and constraints on daughter strings, but this is simply one among myriads of possible combinations of tier projections and constraints. The very marked nature of TTT might actually serve as an explanation for why the *that*-trace effect is attested in very few languages.

In addition, the TSL-2 view also makes it less surprising that we find anti-*that*-trace effects with other movement types (Douglas, 2017):

(7) I met [the woman]$_i$ *(that/who) t_i saw John.

TSL-2 can treat this as a simple variation of the *that*-trace effect such that I) we now operate on a TTT-like variant of the `rel`-tier, where `rel` is the movement feature that extracts head nouns from their relative clause, and II) it is the unpronounced complementizer, not the pronounced one, that bans wh-subjects in its string of tier daughters. A long-standing puzzle reduces to accidental variation across tiers.

4 Opaque and transparent tier buffers

The account of the *that*-trace effect uses two tricks. By setting the number of allowed elements of a specific type to 0, we enforce the absence of those elements in specific daughter strings. But at the same time, additional elements are projected to act as a kind of *tier buffer* that blocks the constraint from applying in specific circumstances. In this section, we will see two additional uses of buffers. Buffers that interrupt licensing conditions give rise to islands (§4.1). Buffers that daisychain licensing conditions give rise to wh-agreement (§4.2).

4.1 Islands: opaque tier buffers

Islands are constituents that are opaque to (certain types of) extraction. A phrase contained within an island may not move to positions outside that island. Some common examples of islands in English are shown in (8).

(8) a. *Well-formed extraction without island*
What$_i$ did John complain that Mary brought t_i to the party?

b. *Adjunct island*
*What$_i$ did John complain because Mary brought t_i to the party?

c. *Whether island*
*What$_i$ did John wonder whether Mary brought t_i to the party?

d. *Complex NP island*
*What$_i$ did John complain about the fact that Mary brought t_i to the party?

e. *Relative clause island*
*What$_i$ did John complain about the person that brought t_i to the party?

The specific configurations that induce island effects vary across languages and even speakers, and so does what types of movement are subject to island effects (see Szabolcsi and Lohndal 2017 and references therein). Hence any good theory of movement must solve multiple puzzles: I) why do island effects exist in the first place, II) why aren't all movement types subject to the same island effects, and III) why aren't island effects uniform across languages and speakers?

The TSL view of movement provides natural answers to all those questions, and it does so without any extra stipulations. Quite simply, islands arise when a tier contains elements that cannot satisfy the need of nodes with $\texttt{f}^+$ for a daughter with $\texttt{f}^-$. Just like seemingly irrelevant nodes on a tier prevent a constraint violation with the *that*-trace effect, with islands such nodes prevent constraint satisfaction.

Consider the dependency tree for sentence (8d) with a complex NP island, as depicted in Fig. 5. The observed island effect is unexpected under the standard tier projection for wh-tiers, which projects all LIs, and only those, that carry $\texttt{wh}^+$ and $\texttt{wh}^-$. As can be seen in Fig. 5 (middle), the resulting tier is well-formed. With the default tier projection, then, the complex NP island effect is entirely unexpected. But there is nothing that prevents English from using a different tier projection where the wh-tier contains not just LIs that carry $\texttt{wh}^+$ or $\texttt{wh}^-$. The wh-tier could just as well contain complex NPs, which are exactly those LIs whose feature annotation starts with $\texttt{C}^+\texttt{N}^-$. The resulting tier, depicted in Fig. 5 (right), now has the two movement nodes separated by $\texttt{fact} :: \texttt{C}^+\texttt{N}^-$. Since this LI carries no movement features at all, $\texttt{f}^+$ is missing a matching $\texttt{f}^-$ among its daughters. This renders the tier ill-formed, and a single ill-formed tier is sufficient to rule out the entire derivation.

Other island constraints similarly reduce to the projection of specific LIs that interrupt licensing relations. Adjunct islands arise whenever adjuncts are projected (in contrast to the *that*-trace effect, here one has to project the adjunct itself instead of the empty adjunction head as extraction from the adjoinee is still permitted). This also includes relative clause islands, which can be analyzed as NP and DP adjuncts. Similarly, *whether* islands are the result of projecting the LI $\texttt{whether} :: \texttt{T}^+\texttt{C}^-$, which once again poses no computational challenges. The same strategy even accounts for subject islands.

(9) *Subject island constraint*
[Which student]$_i$ did [the advisor of t_i] study island constraints?

As long as all subjects carry some $\texttt{nom}^-$ that enforces (overt or covert) subject movement, and as long as $\texttt{nom}^-$ can only occur on subjects, the subject island constraint is the result of projecting every LI with $\texttt{nom}^-$ on every tier. We see, then, that TSL readily accommodates island effects because there is no *a priori* ban against projecting specific LIs onto movement tiers, including those with no movement features at all.

The TSL account also explains why island effects can vary across movement types, and why they aren't universal across languages and speakers. Since every movement tier uses its own tier projection, there is no reason why all tier projections should project the same LIs. By extension, there is also no reason why all languages have to have exactly the same tier projections for every movement type. Note that this even includes exceptions to island constraints, e.g. *Truswell sentences* (Truswell, 2007).

(10) a. * [Which car]$_i$ did John drive Mary crazy while he tried to fix t_i?

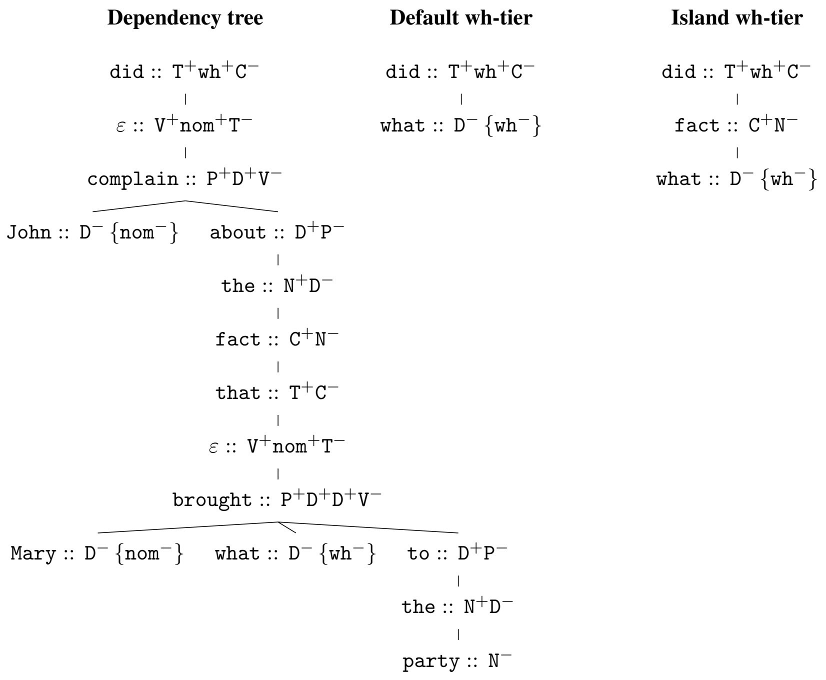


Figure 5: Dependency tree for the complex NP island in (8d) and two choices of wh-tier

b. [Which car]$_i$ did John drive Mary crazy while trying to fix t_i?

Under the plausible assumption that the category feature $\texttt{T}^-$ should actually be split into $\texttt{T}^-_{\mathit{fin}}$ and $\texttt{T}^-_{\mathit{inf}}$ for finite and infinitival TPs, respectively, this split boils down the fact that `while` :: $\texttt{V}^+\texttt{T}^-_{\mathit{fin}}$ projects onto movement tiers whereas `while` :: $\texttt{V}^+\texttt{T}^-_{\mathit{inf}}$ does not. For TSL-2, Truswell sentences are no more remarkable than the fact that *whether* induces islands while *if* does not.

(11) a. * What$_i$ did John wonder whether Mary brought t_i to the party?

b. What$_i$ did John wonder if Mary brought t_i to the party?

Without additional restrictions, TSL allows for free variation in tier projections, and this explains the variability we find across movement types, languages, and speakers.

What more, TSL provides a natural upper bound on the complexity of islands. All of the following are logically feasible island constraints, yet none of them are attested:

(12) a. *Gang-up island effects*
A mover can escape n islands, but not $n+1$.

b. *Configurational island effects*
XP is an island iff it is inside an embedded clause.

c. *Cowardly island effects*
XP is an island iff there are at least n XPs in the same clause.

d. *Narcissist island effects*
XP is an island iff there are no other XPs in the same clause.

e. *Rationed island effects*
At most n phrases per clause can be an island.

f. *Discerning islands*
XP is an island only for movers that contain a PP.

What they all have in common is that the TSL tier projection, which only considers individual nodes/LIs and never their structural context, cannot project nodes in a manner that would match these island effects. A cognitive device that is limited to TSL-2 is simply incapable of expressing such constraints on movement.

4.2 Wh-agreement: Transparent tier buffers

We just saw that islands arise from tier nodes that lack both $\texttt{f}^+$ and $\texttt{f}^-$ and thus interrupt all licensing relations related to those features. But one could

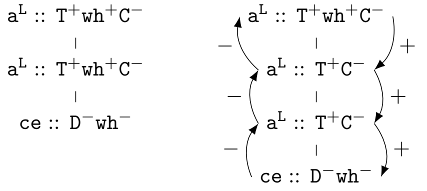


Figure 6: Left: ill-formed tier for (13) where the embedded complementizer carries $\mathtt{wh}^{+}$; Right: tier with licensing relations if *aL* acts as if it had both $\mathtt{wh}^{+}$ and $\mathtt{wh}^{-}$

also imagine the opposite: a node that lacks both features yet acts as if it had both. More than just a technical curiosity, this allows for a novel analysis of wh-agreement in Irish (McCloskey, 1979, 2001) and Chamorro (Chung, 1998), among others.

The example below (McCloskey, 2001, p.94) shows how complementizers in Irish change their phonetic exponent from *go* to *a* or *aL* if a wh-phrase moves across them (the phenomenon also happens with other kinds of movement, but the proposed TSL-2 analysis generalizes to those, too).

(13) Cé aL/*go dúradh léithi a/*go
who C-wh/C was-said with-her C-wh/C
cheannódh é?
would-buy it
'Who was she told would buy it?'

Crucially, this happens to all complementizers along the movement path, no matter how many there are.

The alternation in the first complementizer is easily captured by having two separate lexical entries $\mathtt{a}^{\mathtt{L}} :: \mathtt{T}^{+}\mathtt{wh}^{+}\mathtt{C}^{-}$ and $\mathtt{go} :: \mathtt{T}^{+}\mathtt{C}^{-}$ that differ in the presence of $\mathtt{wh}^{+}$. But the complementizer of the embedded clause cannot carry $\mathtt{wh}^{+}$ — if it did, the wh-tier would be ill-formed (see Fig. 6, left). How, then, can TSL possibly capture the movement-sensitive distribution of *aL* and *go*?

As with the *that*-trace effect and islands, the answer is that projection onto an $\mathtt{f}$-tier need not be limited to LIs with $\mathtt{f}^{+}$ or $\mathtt{f}^{-}$. Suppose that both $\mathtt{a}^{\mathtt{L}} :: \mathtt{T}^{+}\mathtt{C}^{-}$ and $\mathtt{go} :: \mathtt{T}^{+}\mathtt{C}^{-}$ project onto the wh-tier, but exhibit very different types of behavior on this tier. The default complementizer *go* acts like an island for wh-movement: if a clause is headed by *go*, no phrase can wh-move out of it. Hence *go* can never occur along a wh-movement path.

The agreeing complementizer *aL*, on the other hand, behaves as if it carried both $\mathtt{wh}^{+}$ and $\mathtt{wh}^{-}$. Because *aL* acts as if it carried $\mathtt{wh}^{+}$, it requires a negative daughter with $\mathtt{wh}^{-}$. But since *aL* also acts as if it carried $\mathtt{wh}^{-}$, the daughter can be just another instance of *aL*. Eventually, though, the lowest element must be a wh-mover that only carries $\mathtt{wh}^{+}$ and thus puts no requirements on its daughter string. At the same time, the fact that *aL* behaves as if it carried $\mathtt{wh}^{-}$ also means that it must have a mother with $\mathtt{wh}^{+}$. Again this can be another instance of *aL* because *aL* also acts like $\mathtt{wh}^{+}$. But eventually there has to be a node at the very top that only carries $\mathtt{wh}^{+}$ and no $\mathtt{wh}^{-}$ — in other words, a wh-landing site with $\mathtt{wh}^{+}$. Putting all of this together, a sequence of one or more instances of *aL* can only occur sandwiched between $\mathtt{wh}^{+}$ and $\mathtt{wh}^{-}$, i.e. along a wh-movement path.

The TSL-2 account of Irish thus posits a complementizer *go*, which can never occur along a wh-path, and a separate complementizer *aL*, which can occur only along a wh-path. What is usually analyzed as a single complementizer agreeing with a successive-cyclic wh-mover is actually two distinct complementizers that are in complementary distribution due to how they differ in their behavior on the wh-tier.

5 Discussion

We have seen that the TSL-2 characterization not only captures movement in a simple manner, it also accounts for a number of seemingly unrelated phenomena that arise with movement: multiple wh-movement (§3.1), optional movement and expletive constructions (§3.1), *that*-trace effects and anti-*that*-trace effects (§3.2), adjunct islands, complex NP islands, *whether* islands, relative clause islands, subject islands (§4.1), and finally wh-agreement (§4.2). Most importantly, these phenomena require no additional machinery or assumptions. A cognitive system that can handle the TSL-2 dependencies of standard movement has all the computational resources to also handle these phenomena. If we assume free variation in the lexicon and the tier projections, each one of these phenomena is bound to eventually show up in some language. But this is also the shortcoming of the current TSL-2 perspective: languages are much more principled and systematic than the free variation account predicts.

If tier projections vary freely across tiers, languages, and speakers, why then do we find no languages that completely lack the adjunct island constraint? Why do even those languages where relative clauses do not induce island effects still show processing effects that suggest that they are islands (Tutunjian et al., 2017)? Why isn't there a language where the facts for Truswell sentences are exactly the other way around, with infinitival T opaque to extraction whereas finite T allows for it? And why isn't there an analogue of the *that*-trace effect that targets objects instead of subjects? While TSL-2 rules out many unnatural kinds of movement dependencies (cf. (12)), it still allows for any kind of unnatural phenomenon that can be expressed as the projection of a finite subset of the lexicon, no matter how idiosyncratic that subset.

This shows that TSL-2 in its current form still overgenerates and is too lax a restriction on the typology of island constraints. However, the TSL tier projection also provides a natural locus for addressing this overgeneration. What TSL-2 needs is a theory of tier projections. This could come in the form of substantive universals, perhaps coupled with abstract notions like monotonicity (Graf, 2019, 2020; Moradi, 2019, 2020, 2021). Alternatively, there may be restrictions on the relation of tiers to each other, akin to the constraints on harmony tiers identified by Aksënova and Deshmukh (2018). The key point is that while the issue is still open, TSL already furnishes a path towards its solution — in contrast to other analyses of islands, which usually have to add on new machinery to account for unexpected variation rather than pruning down the already predicted typology.

That said, TSL-2 isn't a uniform account of all attested movement constraints, either. As far as I can tell, some conditions on movement simply are beyond the purview of TSL-2, e.g. freezing effects and the Coordinate Structure Constraint. Whether this is an isufficiency of TSL-2 or my own analytical abilities remains to be seen, and it may still be possible to come up with, say, a TSL-3 account of freezing effects. In addition, there are alternative models of subregular dependencies in syntax, foremost constraints on string representations obtained from dependency trees (Graf and Shafiei, 2019; Shafiei and Graf, 2020) and the class of constraints recognizable by sensing tree automata (Graf and De Santo, 2019). Even though these were developed for constraints that do not directly regulate movement, for instance Principle A of binding theory, there is no obvious reason why well-attested conditions on movement cannot come from this class instead. Again the logic is that if these computational resources are already available to handle phenomena like Principle A, it would be surprising if this machinery were never applied to movement. Perhaps, then, TSL-2 covers a large portion of movement, but not the full space, with other subregular classes picking up the slack. Overall, TSL is far from the final word on movement, but it provides a surprisingly versatile starting point that can be refined in various ways (tier projection, going beyond TSL-2) to improve its empirical adequacy.

Conclusion

I have argued that the TSL-2 characterization of Minimalist movement is not a purely mathematical curiosity but an empirically fertile perspective that readily accommodates a large variety of phenomena related to movement. This is a unique conceptual advantage of TSL-2. Whereas other syntactic proposals require additional machinery to go from the basic mechanism of movement to multiple wh-movement, island effects, *that*-trace effects, and wh-agreement, all of them come for free with TSL-2. Any cognitive system capable of movement also has the computational resources to handle these phenomena. Similarly, TSL-2 also predicts that we should never see unnatural things like the *gang-up islands* from (12) because they are not TSL-2, whereas the non-existence of such islands is puzzling under standard Minimalist accounts. Despite all these advantages, TSL-2 is not the final word on movement because it predicts too much variation across movement types, languages, and speakers. Future work should strive to identify abstract properties of tier projections that separate natural from unnatural movement phenomena.

Acknowledgments

The work reported in this paper was supported by the National Science Foundation under Grant No. BCS-1845344. I thank the three anonymous reviewers for their detailed comments. Reviewer 3 in particular provided numerous thought-provoking observations and suggestions, many of which will find their way into follow-up work with less severe space constraints.

References

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

Željko Bošković. 2002. On multiple wh-fronting. *Linguistic Inquiry*, 33:351–383.

Marguerite Browning. 1996. CP recursion and *that*-t effects. *Linguistic Inquiry*, 27:237–255.

Sandra Chung. 1998. *The Design of Agreement: Evidence from Chamorro*. Chicago University Press, Chicago.

Jamie Douglas. 2017. Unifying the *that*-trace and anti-*that*-trace effects. *Glossa*, 2:1–28.

Werner Frey and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in Minimalist grammars. In *Proceedings of the Conference on Formal Grammar*, pages 41–52.

Thomas Graf. 2014. Models of adjunction in Minimalist grammars. In *Formal Grammar 2014*, volume 8612 of *Lecture Notes in Computer Science*, pages 52–68, Heidelberg. Springer.

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

Thomas Graf. 2019. Monotonicity as an effective theory of morphosyntactic variation. *Journal of Language Modelling*, 7:3–47.

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, Alëna Aksënova, and Aniello De Santo. 2016. A single movement normal form for Minimalist grammars. In *Formal Grammar: 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016*, pages 200–215, Berlin, Heidelberg. Springer.

Thomas Graf and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 12–26, Toronto, Canada. Association for Computational Linguistics.

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

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

Jeffrey Heinz, 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.

Tim Hunter. 2015. Deconstructing merge and move to make room for adjunction. *Syntax*, 18:266–319.

James McCloskey. 1979. *Transformational Syntax and Model Theoretic Semantics: A Case Study in Modern Irish*. Reidel, Dordrecht.

James McCloskey. 2001. The morphosyntax of wh-extraction in Irish. *Journal of Linguistics*, 37:67–100.

Sedigheh Moradi. 2019. *ABA generalizes to monotonicity. In *Proceedings of NELS 49*, volume 2.

Sedigheh Moradi. 2020. Morphosyntactic patterns follow monotonic mappings. In *Monotonicity in Logic and Language*, pages 147–165, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sedigheh Moradi. 2021. A formal restriction on gender resolution. In *All Things Morphology: Its Independence and Its Interfaces*, pages 41–54. John Benjamins, Amsterdam.

Nazila Shafiei and Thomas Graf. 2020. The subregular complexity of syntactic islands. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*, pages 272–281.

Edward P. Stabler. 1997. Derivational Minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.

Edward P. Stabler. 2011. Computational perspectives on Minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.

Anna Szabolcsi and Terje Lohndal. 2017. Strong vs. weak islands. In Martin Everaert and H.C. Riemsdijk, editors, *The Wiley Blackwell Companion to Syntax, Second Edition*, pages 479–532.

Robert Truswell. 2007. Extraction from adjuncts and the structure of events. *Lingua*, 117:1355–1377.

Damon Tutunjian, Fredrik Heinat, Eva Klingvall, and Anna-Lena Wiklund. 2017. Processing relative clause extractions in Swedish. *Frontiers in Psychology*, 8:2118.

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

Analysis of Language Change in Collaborative Instruction Following

Anna Effenberger[1], **Eva Yan**[2]*, **Rhia Singh**[2]*, **Alane Suhr**[1], and **Yoav Artzi**[1]
[1]Cornell University
[2]City University of New York
ae347@cornell.edu eyan0749@gmail.com
rhia.singh@macaulay.cuny.edu {suhr, yoav}@cs.cornell.edu

1 Introduction

Community language change in situated collaborative task-oriented scenarios has been studied with focus on reference games (Krauss and Weinheimer, 1964; Clark and Wilkes-Gibbs, 1986; Hawkins et al., 2017, 2020a,b), where two participants coordinate using language to select to a single item from a set of available items. These studies found that utility-maximizing participants trade surface-form linguistic complexity with established norms, as the familiarity and expertise of the interaction partners increase. In practice, this emerges as a reduction in utterance length and vocabulary size.

We study the generality of these observations by analyzing language change in a collaborative instructional task, where instructors can specify multiple goals within a single instruction to increase their utility. This option, not present in reference games, creates competing incentives: increasing utility by issuing more goals in a single instruction versus decreasing language effort by utilizing established norms (e.g., by shortening instructions).

We use the CEREALBAR game environment and its accompanying dataset (Figure 1; Suhr et al., 2019). CEREALBAR is a two-player, collaborative language game where players work together to collect sets of matching cards. A leader plans which cards to include in the next set, and writes instructions to a follower describing tasks to accomplish. In contrast to reference games (Krauss and Weinheimer, 1964), the language in CEREALBAR is primarily instructional rather than referential, and the game allows players to complete a dynamic number of tasks per instruction and game.

Similar to previous studies, we observe language change over time along the same dimensions. But, unlike in reference games, we observe utterance-level linguistic complexity increases. Our study illustrates that the formation of common ground among interaction participants does not necessarily reduce language complexity, and may even come with an increase in complexity. Understanding how humans use language to collaborate in settings with flexible utility is key to building natural language systems that effectively collaborate with users over time. Our analysis code can be found at github.com/lil-lab/CB-analysis.

Decile 1: *get the card in front*

Decile 5: *Collect the green square card in front of you.*

Decile 10: *turn around on the trail, go straight and get 2 green circles, continue straight on the trail to the right side of the glacier and get 1 black triangle.*

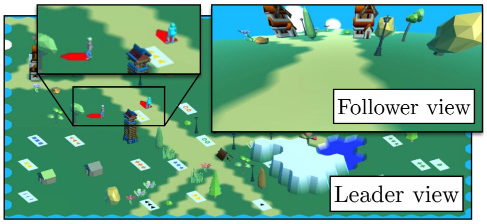


Figure 1: Leader instructions in CEREALBAR from games played at the beginning (Decile 1), middle (Decile 5), and end (Decile 10) of the community life. The differences between the instructions illustrate the linguistic change observed in the data. The instruction from Decile 10 is paired with a snapshot from the game as the follower begins to execute it. The leader (left) and follower (right) are highlighted in the center-left of the leader's view of the game, and the top right shows the follower's first-person view of the environment.

2 Scenario and Data Overview

We use the CEREALBAR game and accompanying dataset (Suhr et al., 2019) in our analysis. CEREALBAR is a collaborative, two-player game, where a leader and a follower collect matching sets of cards by moving in an environment. The game is turn-based, and each player has a limited number of steps per turn. The leader both collects cards and instructs the follower using natural language.[1] The

[*] Equal contribution.

[1] All utterances are in English.

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 194-202.
Held on-line February 7-9, 2022

	Mean	Median	Max
Interaction Score (# Card Sets)	8.8	10.0	19
# Instructions / Interaction	22.0	26.0	41
# Tokens / Instruction	14.4	13.0	55
Vocabulary Size		3,499	
Total # Instructions		17,524	

Table 1: Statistics of analyzed data.

follower executes leader instructions. The players' abilities differ: the leader observes the complete environment and plans sets to collect; the follower only observes what is ahead, but has more steps per turn. For each set made, players receive one point and additional turns, allowing them to complete more sets. Success requires the players to collaborate via natural language: the leader must write informative instructions to the follower, and the follower must efficiently follow these instructions. Figure 1 shows a snapshot of the game.

The CEREALBAR dataset contains 1,202 human-human game interactions collected over the course of four months. Workers were randomly assigned as leader or follower for each interaction. The collection process created a Wizard-of-Oz setup: the system user, as the leader, provides instructions and acts in the world, and the human follower is a wizard, executing instructions to emulate the desired system behavior. We only use interactions from the training split for our analysis. We prune interactions by inexperienced workers, as classified when the data was collected, to focus on the impact of experience.[2] In total, we consider 795 interactions. Table 1 provides basic statistics of the data we use. Suhr et al. (2019) used these data to train models, while we study how the language changes.

3 Data Analysis

To analyze trends over the data collection period, we split the data chronologically into 10 deciles of roughly equal size (79 or 80 interactions). An average of 40 workers participated in each decile (Figure 2, left). The community stabilized after Decile 4, as worker recruitment slowed and the community was split by expertise.[3]

Interaction goals are increasingly achieved over time. Mean score per game increases from 3.8 to 12.3 ($p < 0.0001$) (Figure 2, right).[4] Execution efficiency and game expertise also improve.[5] Our focus is how leader language - the sole communication conduit - changes to enable these gains.

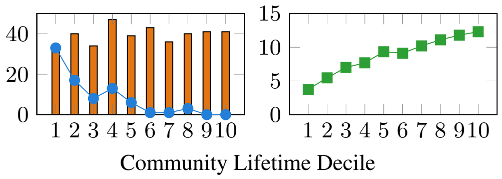


Figure 2: Community size (left) and mean game score (right) over deciles of community lifetime. On the left, the bars show total active players and the curve shows only the number of new players that joined per decile.

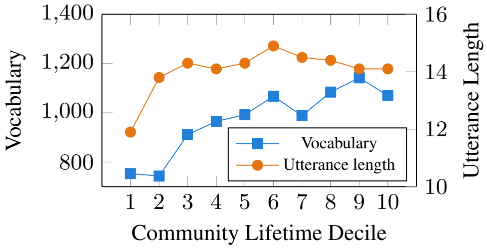


Figure 3: Vocabulary and utterance length over deciles.

We design our analysis to be as similar as possible to existing work on reference games (Hawkins et al., 2020a), which shows that certain language aspects are simplified as community conventions form. CEREALBAR allows for a different realization of common ground development than previously studied reference games, and we observe trends that are in contrast to this line of prior work.

Instruction Length and Vocabulary Mean[6] instruction length increases from 11.9 to 14.1 tokens[7] ($p < 0.0001$) over time, while vocabulary size increases from 752 to 1,070 unique tokens (Figure 3). This contrasts with reference games, where utterance length and vocabulary size reduce (Clark and Wilkes-Gibbs, 1986; Hawkins et al., 2017). Some of the words added more specifically describe props or movements. However, the overall trend is relatively complex, and identifying clear patterns likely requires a more targeted scenario design.

Syntactic Complexity We analyze syntactic trends using parts-of-speech (POS) tags and dependency trees.[8] We do not observe a significant difference in usage of closed- and open-class POS tags, as seen in reference games (Hawkins et al., 2017). We observe change in the relative use

[2] Appendix B.1 describes this pruning process.

[3] Appendix B.2 provides decile details.

[4] We use a two-sided t test at $\alpha = 0.05$ for all calculations of significance when comparing means.

[5] Appendix C.1 details this improvement.

[6] All means over instructions are first computed within each game, then across games. This weighs all games equally, rather than upweighing longer, higher-scoring games.

[7] We use NLTK for tokenization, lowercase all tokens, and use the `autocorrect` library for typo correction.

[8] We use spaCy (Honnibal and Montani, 2017) for POS tagging and dependency parsing.

of verbs, nouns, conjunctions, determiners, and numerals.[9] Notably, the proportion of conjunctions of all tokens increases from 0.060 to 0.067 ($p = 0.0026$).[10] The proportion of instructions that contain a conjunction also increases from 0.0495 to 0.0707 ($p = 0.0113$). Qualitatively, this accompanies an increased use of ordered sentential conjunctions, often to specify multiple tasks in a single utterance (e.g., *once you get that card, turn around and go left and get the 1 green circle card*).

We compute three measures of syntactic complexity using dependency trees (Xu and Reitter, 2016): (a) maximum depth: the longest path from root to a leaf; (b) maximum width: the maximum out-degree of any node; and (c) average branching factor: the average out-degree of non-leaf nodes.[11] We normalize to control for utterance length. Figure 4 shows these statistics over time. Maximum width and branching factor increased from 0.941 to 0.987 ($p = 0.0483$) and from 0.934 to 1.00 ($p = 0.0051$), indicating increased descriptiveness. Maximum depth did not significantly change, indicating embedded clause use proportional to length, as expected when increasingly combining instructions with conjunctions. We observe similar trends when comparing these statistics between low- and high-scoring games (Appendix C.2).

Overall, our syntactic analysis shows an increase in language complexity is required to describe more tasks within a single instruction. We do not observe a gradual drop of redundant modifiers and descriptors (Hawkins et al., 2017). This may be because potential referents do not pose as much ambiguity as the abstract shapes often used in reference games (Clark and Wilkes-Gibbs, 1986).

Changes in References We see no significant development of niche idioms, in contrast to reference games with abstract shapes (Hawkins et al., 2020a). This is likely due to concreteness and familiarity of the referents in CEREALBAR, allowing players to rely on common background knowledge with little ambiguity. We observe change in the relative frequency of references to specific objects over time. We consider seven object classes: building, road, foliage, rock, ice, water, and light.[12] The proportion of instructions containing a reference to ice, light, and buildings increase from 0.006 to 0.022 ($p = 0.0006$), from 0.015 to 0.027 ($p = 0.0188$), and from 0.056 to 0.073 ($p = 0.0436$). The ratios of other references are stable. Leaders likely choose references to balance informativity and effort. Foliage objects are common and require more effort to differentiate, while buildings and ice clearly vary. Lights, though common, were often referred to with other objects to clarify location.

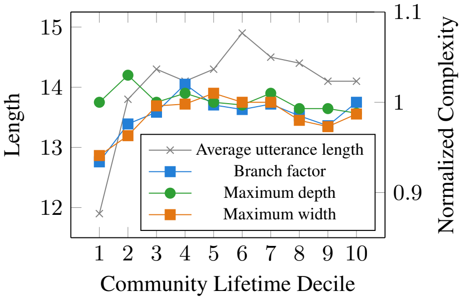


Figure 4: Average syntactic branching factor, maximum depth, and maximum width across deciles. We also plot the mean utterance length for reference.

Language Effort Leaders in CEREALBAR mainly instruct followers to complete card events to ultimately select valid card sets. We measure language effort with respect to this objective as the number of tokens and instructions per card event (Figure 5). This notion of effort is similar to utterance cost in speaker-listener pragmatic models (Goodman and Frank, 2016). The number of instructions per card event decreases from 0.879 to 0.783 ($p = 0.0102$), indicating leaders effectively pack more tasks into fewer instructions – often multiple card events into one instruction in later deciles (Figure 1). This change correlates with structural changes. For example, conjunctions are useful to pack more tasks into single instructions; the correlation across deciles between the proportion of instructions containing a conjunction and the number of instructions per card event is $r = -0.8243$. The high negative correlation indicates that the change in conjunction use aligns with the increase in goals (i.e., cards to select) packed per instruction. The number of tokens per card event initially increases from 9.9 to 11.8, then decreases to 10.7. This may be because, initially, followers require more verbose instructions and leaders experiment with the level of description, but as conventions form, this verbosity is less needed to understand instructions.

The reduction in the number of tokens per goal later on corresponds to the reduction in utterance length observed in reference games (Hawkins

[9] Appendix C.2 provides details.

[10] We use a one-sided z test at $\alpha = 0.05$ for calculations of significance when comparing proportions.

[11] We further explain the syntactic measures and provide example instructions for illustration in Appendix C.2.

[12] Appendix C.3 describes this classification process.

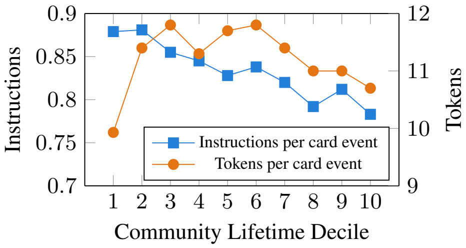


Figure 5: The number of instructions and tokens required for a card event over deciles. Analysis considers only instructions marked complete by the follower.

et al., 2017), although it is manifested differently as the overall surface-form is not simplified (i.e., via shorter utterances), unlike in reference games. Given the opportunity to increase utility, leaders choose to take advantage of followers' increased expertise and efficiency by using more complex language to pack more goals into each instruction.

4 Discussion and Related Work

The CEREALBAR scenario is related to reference games (Krauss and Weinheimer, 1964; Clark and Wilkes-Gibbs, 1986; Hawkins et al., 2017; Monroe et al., 2017; He et al., 2017; Udagawa and Aizawa, 2019; Haber et al., 2019), which require two players to agree on a single referent from a set via dialogue. CEREALBAR differs in several ways. It allows only unidirectional language communication,[13] and utterances in CEREALBAR are instructions specifying desired follower behavior with any number of tasks to complete (i.e., with flexible utility), not a description of a single target referent.

These differences lead to different language dynamics. In reference games, Hawkins et al. (2020a) observed the development of specialized reference phrases for ambiguous shapes, which allows players to reduce their utterances' length and syntactic complexity. Given that CEREALBAR objects are generally unambiguous and familiar, players do not begin with overly verbose references, and have less potential for reduction to more concise references. In contrast, we observe increased instruction length and complexity. Leaders issue an increasing number of tasks to the follower per instruction, utilizing the flexibility afforded by CEREALBAR's design. This less constrained scenario better reflects real-life collaborations, where participants complete many tasks to achieve complex goals.

Our observations show the competing effects of cost-minimization and utility-maximization. The formation of common ground and expectations on partners' behavior enables leaders to use language differently to convey more information-dense instructions to optimize game performance. This is aligned with the expectation of better communication grounding between community members in Clark and Marshall (1981), and with how grounding in Clark and Wilkes-Gibbs (1986) manifests as reduced complexity when utterance utility is fixed. Because there are conflicting forces at work in CEREALBAR, common ground is realized differently.

The most related setup to CEREALBAR is the Cards task (Djalali et al., 2012; Potts, 2012), where two players collect a single set of cards. It uses four static environments and studies dialogue, not instructions. Djalali et al. (2011) showed Cards players increase the interaction complexity by developing a rich common ground, including terms for the fixed board locations. This is less likely with the randomly generated CEREALBAR environments. Utterances in Cards also become shorter, potentially due to the predefined number of goals.

The language dynamics observed in CEREALBAR contrast with those previously observed in reference games, providing evidence that gradual formation of common ground among interaction participants does not necessarily result in reduced complexity of sentences, and may even result in increased complexity. Our conclusions do not void nor mutually exclude previous work, but illustrate the complexity of language change over time in a community. An important direction for future work is controlled studies to observe the effects of scenario design on the interaction between the development of common ground and language change.

Acknowledgments

This research was supported by NSF under grants No. 1750499, 1750499-REU, and DGE-1650441. It also received support from a Google Focused Award, the Break Through Tech summer internship program, and a Facebook PhD Fellowship. We thank Chris Potts and Robert Hawkins for early discussions that initiated this analysis; and Ge Gao and Forrest Davis for their comments.

[13] Language change in unidirectional reference games was also studied by Krauss and Weinheimer (1966), who found that when task-completion feedback is provided, references simplify over time.

References

Herbert H. Clark and Catherine R. Marshall. 1981. Definite knowledge and mutual knowledge. *Elements of discourse understanding*, pages 10–63.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Alex Djalali, David Clausen, Sven Lauer, Karl Schultz, and Christopher Potts. 2011. Modeling expert effects and common ground using questions under discussion. In *AAAI Fall Symposium: Building Representations of Common Ground with Intelligent Agents*.

Alex Djalali, Sven Lauer, and Christopher Potts. 2012. Corpus evidence for preference-driven interpretation. In *Logic, Language and Meaning*.

Noah D. Goodman and Michael C. Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20:818–829.

Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. 2019. The PhotoBook dataset: Building common ground through visually-grounded dialogue. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Robert X. D. Hawkins, Michael C. Frank, and Noah D. Goodman. 2020a. Characterizing the dynamics of learning in repeated reference games. *Cognitive science*, 44 6:e12845.

Robert X. D. Hawkins, Mike Frank, and Noah D. Goodman. 2017. Convention-formation in iterated reference games. In *Cognitive Science*.

Robert X. D. Hawkins, Noah D. Goodman, A. Goldberg, and T. Griffiths. 2020b. Generalizing meanings from partners to populations: Hierarchical inference supports convention formation on networks. In *Proceedings of the Annual Conference of the Cognitive Science Society*.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Robert M. Krauss and Sidney Weinheimer. 1964. Changes in reference phrases as a function of frequency of usage in social interaction: a preliminary study. *Psychonomic Science*, 1:113–114.

Robert M. Krauss and Sidney Weinheimer. 1966. Concurrent feedback, confirmation, and the encoding of referents in verbal communication. *Journal of personality and social psychology*, 4 3:343–6.

Will Monroe, Robert X.D. Hawkins, Noah D. Goodman, and Christopher Potts. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338.

Christopher Potts. 2012. Goal-driven answers in the Cards dialogue corpus. In *Proceedings of the West Coast Conference on Formal Linguistics*, pages 1–20.

Alane Suhr, Claudia Yan, Jack Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Takuma Udagawa and Akiko Aizawa. 2019. A natural language corpus of common grounding under continuous and partially-observable context. In *Proceedings of the Conference on Artificial Intelligence*.

Yang Xu and David Reitter. 2016. Convergence of syntactic complexity in conversation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

A Reproducibility Checklist Details

All computation was done on a personal laptop. The CerealBar data was acquired from `https://github.com/lil-lab/cerealbar`.

B Data Details

B.1 Selection of Interactions for Analysis

The data we use was not collected specifically for this analysis, but during data collection for model development by Suhr et al. (2019). We use 795 of the 960 interactions in the original training split of the data for our analysis, pruning the rest to avoid games that include inexperienced players later in the community's life. This prevents the language of novice workers from affecting our analysis after the more experienced community had stabilized, which would potentially suppress convention formation trends observed in existing literature about reference games (Hawkins et al., 2020a). During the original data collection process, after 367 of the 960 total training interactions were collected, the community was split into junior and senior workers. Junior workers became senior upon gaining adequate experience. A junior worker could request to be moved to the senior pool after they had played at least one game as a follower and at least one game as a leader where they earned at least one point with their partner, and they seemed to be following the game rules. Workers who performed well before the split were included in the senior pool. We do not consider games from the junior pool.

B.2 Decile Details

All deciles span a relatively short period of time except the sixth decile, which includes a pause in data collection (Table 2). The pause did not significantly effect community membership or performance. Figure 6 shows the number of instructions per decile, distinguished by complete and incomplete instructions. Incomplete instructions occur at the end of an interaction, when there is insufficient time or turns to complete the instruction. Figure 7 shows mean interaction length in each decile. Figure 8 shows follower path lengths per instruction across each decile.

C Additional Analysis Details

C.1 Interaction Performance

Several measures demonstrate an increase in player expertise. We analyze interaction performance through how many moves are taken per each instruction, the occurrence of de-selection card events, and instruction queuing behavior. We find that followers become better at following instructions and leaders at creating efficient plans.

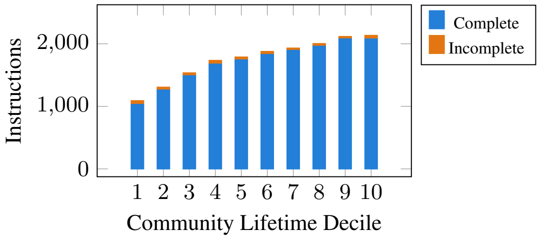


Figure 6: The number of instructions for each decile, distinguished by whether they were marked as complete by the follower.

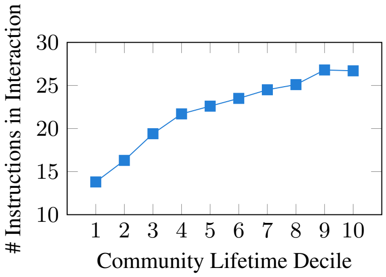


Figure 7: Mean interaction length, measured by the number of instructions, in each decile. We include incomplete instructions in these counts.

Optimal Path Length Deviations We measure how leaders utilize the larger number of steps per turn available to followers through the length of the shortest possible path corresponding to each instruction. We compute this shortest path using the observed start and end positions of the human follower, ensuring that the path avoids obstacles and completes card events completed by the original follower. The mean length of the shortest path per instruction increases over the community lifetime from 6.66 to 7.97 moves ($p < 0.0001$). This corresponds to the increase we observe in the number of goals described in each instruction, which likely requires more steps.

Concurrently, we see improvements in follower instruction execution, measured through the excess moves taken by follower: the difference between the number of moves the follower took and the shortest possible path corresponding to each completed instruction. Over time, the number of excess steps compared to the shortest paths decreased from 3.67 to 2.36 moves ($p < 0.0001$). Figure 9 visualizes this increase in average optimal path length per instruction and decrease in moves taken in ex-

Decile	Game IDs	Lower Time Limit	Upper Time Limit	Time (Days)
1	1-79	2019-01-27 20:05:00 UTC	2019-02-02 15:39:00 UTC	5.815278
2	80-159	2019-02-02 15:39:00 UTC	2019-02-02 20:24:00 UTC	0.197917
3	160-238	2019-02-02 20:24:00 UTC	2019-02-03 00:25:00 UTC	0.167361
4	239-318	2019-02-03 00:25:00 UTC	2019-02-04 00:15:00 UTC	0.993055
5	319-397	2019-02-04 00:15:00 UTC	2019-02-04 03:09:00 UTC	0.120833
6	398-477	2019-02-04 03:09:00 UTC	2019-04-15 19:27:00 UTC	70.6375
7	478-556	2019-04-15 19:27:00 UTC	2019-04-15 23:44:00 UTC	0.178472
8	557-636	2019-04-15 23:44:00 UTC	2019-04-16 20:06:00 UTC	0.848611
9	637-715	2019-04-16 20:06:00 UTC	2019-04-16 22:50:00 UTC	0.113889
10	716-795	2019-04-16 22:50:00 UTC	2019-04-17 03:43:00 UTC	0.203472

Table 2: Time limits of the division into deciles. The last column is the total amount of time elapsed during a decile. All lower time limits are inclusive. All upper time limits are exclusive, except the last one, which is inclusive.

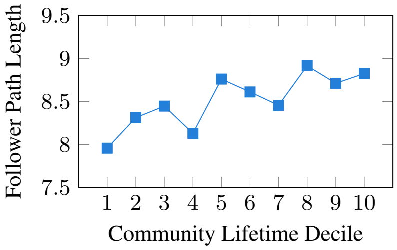


Figure 8: Mean length of observed follower paths for complete instructions in each decile. We measure length in the number of steps recorded per instruction.

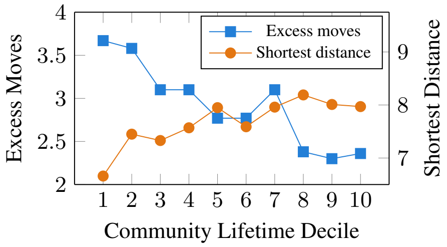


Figure 9: Excess follower moves and shortest possible distance per leader instruction.

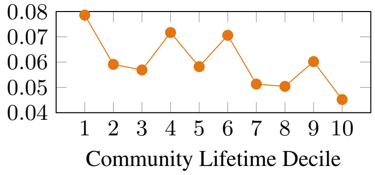


Figure 10: Proportion of all card events, initiated by both followers and leaders, that were de-selections.

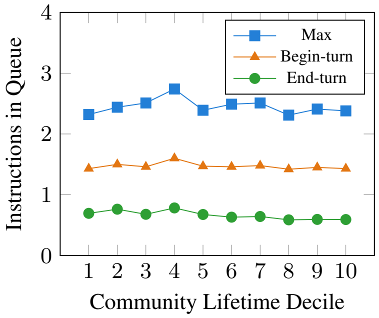


Figure 11: Instruction-queuing behavior over time.

cess of this optimal path. The reduction in excess moves is especially notable given the increase in the moves required per instruction, indicating the absolute decrease observed is due to an even higher decrease in the probability of follower errors.

Card De-selections We also study the occurrence of card de-selections, which often reflect error correction. In ideal gameplay, no de-selection events should be observed, as they require additional steps and only correct for a mistakenly selected card not to be part of the current target set. We observe that player errors decrease: the proportion of card events (the selection or de-selection of a single card) that are de-selections decreases from 7.86% to 4.52% ($p = 0.0018$). Figure 10 shows the percentage of card events initiated by either player that are de-selections.

Instruction Queuing The CerealBar setup allows a leader to plan ahead by queuing multiple instructions to the follower at a time. For example, to efficiently use all of the follower's moves, a leader may send two instructions: one which tells them to complete the set, and another that tells them to move towards a card which will make up the next set. A larger queue indicates longer-term leader planning. Alternatively, the leader could include the additional information in one instruction without queuing more instructions. We analyze this queuing behavior as a potential alternative explanation: the leaders may improve how they relay information with better planning, rather than changing the content of their instructions.

We measure the size of the queue at the beginning and end of follower turns, and the maximum queue size reached during a game. Figure 11 shows queue statistics over time. Begin-turn queue size directly measures how leaders plan via queuing instructions, as no instructions are queued during the follower's turn. Begin-turn and maximum queue size did not change significantly over

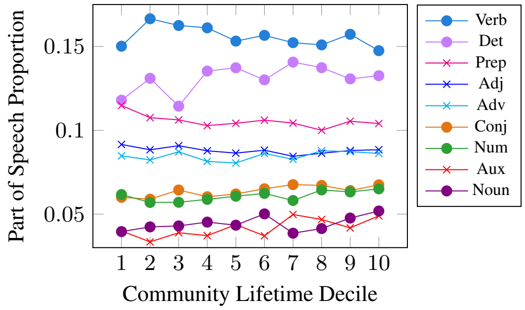


Figure 12: Ratio of language that is a specified part of speech over time. Parts of speech of particular interest are plotted with filled markers.

Dep = 0.83, Wid = 0.93, Bch = 0.83
turn to the left to see one yellow sqaure

Dep = 1.14, Wid = 1.03, Bch = 0.96
go forward one and to your left is orange

Dep = 1.58, Wid = 0.66, Bch = 0.65
take the green card with 3 symbols in front of you

Dep = 0.79, Wid = 1.26, Bch = 1.01
Head straight towards the blue plus card, but don't pick it up. Continue past it, on the left of it.

Figure 13: Selected instructions to illustrate the different measures of complexity, namely: maximum depth (dep), maximum width (wid), and average branching factor (bch). All measures normalized for length.

time. This relative stability indicates that game play improvements were not primarily due to leaders planning ahead across separate instructions; rather, they can be attributed more to the changes of language within instructions. End-turn queue size sampling indicates the efficiency of player collaboration. From the first to last decile, the average end-turn queue size decreases from 0.694 to 0.592 instructions. This indicates that followers become more efficient over time, completing more instructions per turn. This aligns with our analysis of follower efficiency (Section C.1 and Figure 9).

C.2 Syntactic Complexity

Part-of-Speech Analysis To compute the ratio of POS use, we treat each decile of community life as a bag of words, dividing the total tag count of each POS by the total token count in each decile. In our analysis, we combine the spaCy tags ⟨sconj⟩ (subordinating conjunction) and ⟨cconj⟩ (coordinating conjunction) into one conjunction class, and the tags nouns and proper nouns into one noun class. Figure 12 shows the proportion of the nine most common POS tags used in CEREALBAR instructions: verbs, determiners, prepositions, adjectives, adverbs, conjunctions, numerals, auxiliary verbs, and nouns.

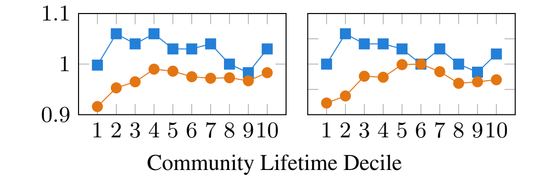


Figure 14: Average dependency branching factor (left) and maximum width (right) over deciles split to games that were above (blue) / below (orange) that decile's median game score.

Syntactic Complexity Analysis For each utterance, we measure the branching factor, maximum width, and maximum depth of its dependency parse. Dependency tree depth indicates how many embedded clauses the utterance has, whereas width-related measures indicate how many modifiers are stacked in one sub-tree. Intuitively, increased width-related metrics indicate more descriptive utterances, whereas increased depth indicates more compounded phrases. Figure 13 provide examples to illustrate these differences.

We normalize these measures by the utterance length following Xu and Reitter (2016). Formally, let X_n be the set of all utterances in our data with a length of n tokens. The average of metric S (e.g., maximum width) across all utterances of length n in our data is:

$$\overline{S}(n) = \frac{1}{|X_n|} \sum_{x \in X_n} s(x) \quad . \tag{1}$$

For each utterance x with length n, we compute the normalized measure for the utterance:

$$s'(x) = \frac{s(x)}{\overline{S}(n)} \quad . \tag{2}$$

Syntactic Complexity and Score We observe similar trends when measuring these statistics when comparing low- and high-scoring games (Figure 14). Higher scoring games had, on average, instructions with significantly higher width and branching factor. In Decile 1, language in games scoring 1 point and 16 points had an average normalized branch factor of 0.915 and 1.02. However, games in the lower 50% of scores showed a higher increase in syntactic complexity over time.

C.3 Reference Change

We divide environmental objects in the CerealBar game into six classes: road, foliage, building, water, rock, ice, and light class objects. We use regular

Class	Keywords
Road	*road*, *fork*, *path*, *intersect*, *trail*, *crossroad*, *crosspath*, *walkway*
Foliage	*palm*, *flower*, *tree*, *shrub*, *grass*, *pine*, *bush*, *grove*, *plant*, *conif*, *field*, *foliag*, *wasteland*, *forest*, *clearing*, *patch*, *lawn*
Building	*tower*, *building*, *house*, *tent*, *barn*, *fort*, *doghouse*, *hut*, *village*, *cabin*, *shack*, *structure*, *shed*, *tower*
Water	*lake*, *pond*, *water*, *sea*, *river*, *coast*, *island*, *shore*
Rock	*rock*, *cliff*, *boulder*, *mountain*, *hill*, *log*, *stone*
Ice	*glacier*, *ice*, *iceberg*
Light	*post*, *lamp*, *pole*, *light*

Table 3: Reference class keywords

expressions to automate if an utterance refers to a class of objects, defined by if it contains at least one of the class keywords in Table 3.

When classifying arguments, BERT doesn't care about word order . . . except when it matters

Isabel Papadimitriou
Stanford University
isabelvp@stanford.edu

Richard Futrell
University of California, Irvine
rfutrell@uci.edu

Kyle Mahowald
University of Texas at Austin
mahowald@utexas.edu

While contextual embedding models are often praised for capturing rich grammatical structure, a spate of recent work has shown that they are surprisingly invariant to scrambling word order (Sinha et al., 2021; Hessel and Schofield, 2021; Pham et al., 2020; Gupta et al., 2021; O'Connor and Andreas, 2021) and that grammatical knowledge like part of speech, often attributed to contextual embeddings, is actually also captured by fixed embeddings (Pimentel et al., 2020). These results point to a puzzle: how can syntactic contextual information be important for language understanding when the words themselves, not their order, are what matter?

We argue that this apparent paradox arises because of the redundant structure of language itself. Lexical distributional information alone captures a great deal of meaning (Erk, 2012; Mitchell and Lapata, 2010), and the local coherence of words is crucial for constructing meaning in both humans (Mollica et al., 2020) and machines (Clouatre et al., 2021). Viewing this redundancy from the perspective of **grammatical role** (whether a noun is the subject or the object of a clause), most clauses are **prototypical**: in a sentence like "the chef cut the onion", the grammatical roles of *chef* and *onion* are clear to humans from the words alone, without word order or context (Futrell et al., 2019, experiments in English and Russian). This means syntactic word order is redundant with lexical semantics. Whether hand-constructed or corpus-based, most studies probing contextual representations have used prototypical sentences as input, where syntactic context does not have much information to contribute to core meaning beyond the words themselves.

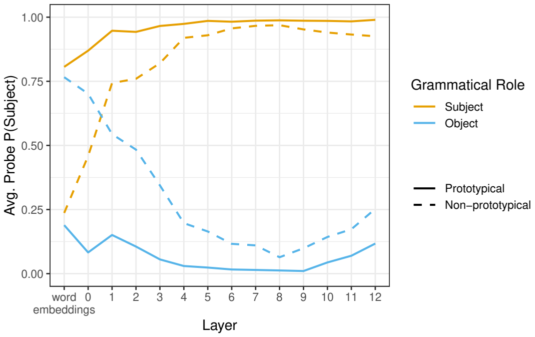


Figure 1: Probabilities of probes trained to differentiate subjects from objects in BERT embeddings. We separate our evaluation examples by prototypicality: whether the grammatical role is what we would expect given the word out of context. The majority of natural examples are prototypical (solid lines), and so if we average all cases we cannot see that grammatical information is gradually acquired in the first half of the network for cases where lexical information is non-prototypical.

Yet human language can use syntax to deviate from the expectations generated by lexical items alone: we can also understand the absurd meaning of a rare **non-prototypical** sentence like "The onion cut the chef" (Gibson et al., 2013). We argue that a linguistically informed understanding of the role of word order information in human language can illuminate the role of context in contextual embedding models.

Our primary experimental method consists of training probing classifiers on the hidden layer embeddings of English BERT (a separate classifier for each layer), to identify whether a noun token is the subject or the object of a transitive sentence. Our experiments rest on comparing the **grammatical subjecthood** of a noun in a sentence (as annotated in the English GUM and EWT UD treebanks) with the **type-level (fixed embedding) subjecthood** prediction: what role we would expect that noun to have if we did not have any context. This allows us to separate prototypical sentences, where the subjects are animate, agentive words (eg, "The man held the umbrella") from

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 203-205.
Held on-line February 7-9, 2022

non-prototypical sentences where the subjects are words generally more likely to be objects (eg, “The umbrella protected the man”). We also perform word order ablations to further understand how structural information arises in the embeddings of non-prototypical examples.

Result 1: Subjecthood is recovered at different layers of BERT, depending on context Prototypical and non-prototypical subjects differ in their probing behaviors between layers (the solid lines in Figure 1). For prototypical subjects, syntactic information is conflated with type-level information and so probe accuracy is high starting from layer 0 (word embeddings + position embeddings), and this stays consistent throughout the network. However, when we look at non-prototypical subjects, we see that the embeddings from layer to layer have very different grammatical encodings, with type-level semantics dominating in the early layers and more general syntactic knowledge only becoming extractable later. Since prototypical subjects dominate in frequency in any corpus, if we were to take the average of all examples, we would see a very moderate change in accuracy through layers. Separating out non-prototypical examples clearly illustrates how the syntax of a phrase arises independently from type-level information through transformer layers, while also showcasing the importance of lexical semantics in forming early layer embedding spaces.

Result 2: Word-order information influences grammatical embedding In our first set of results, we do not differentiate between grammatical information that comes from syntactic word order, and that which is derived from distributional co-occurence information. To address this confound, we repeat our experiment with sentence pairs of the type “The chef cut the onion” → “The onion cut the chef”, where we take a sentence from the treebank data and swap the positions of the subject and the object, thus swapping their roles. As shown in Figure 2, it is possible to extract accurate subjecthood information from these examples, which consist of the same words in the same distributional context. This shows how grammatical-semantic information in embeddings is in fact independently influenced by syntactic word order.

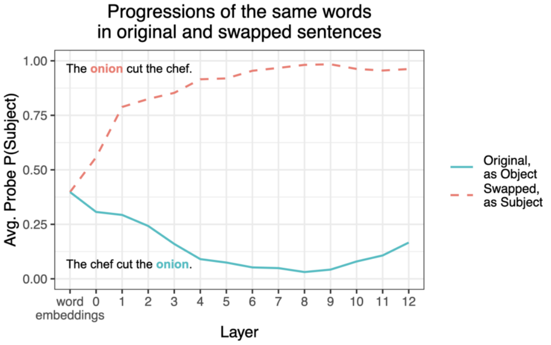


Figure 2: Probe probabilities for the same words when they are the object of an original treebank sentence (blue line) versus being the subject of that sentence after manual swapping (dashed red line). *The same words in the same distributional contexts* are clearly differentiated throughout contextualization in BERT layers, due to the impact of syntactic word order.

Result 3: Fine-grained position information matters for the difficult cases A question still remains: does grammatiacal subjecthood embedding stem from the fine-grained ways in which word order influences syntax in English, or from heuristics based on general primacy (whether a word is earlier or later in a sentence)? To further investigate this, we train and test probes on treebank sentences where we randomly scramble the local word order so that no word moves more than 2 slots, and so general primacy is preserved. For non-prototypical cases, probes trained on these locally shuffled sentences cannot fare better than chance (prototypical cases can be classified with relatively high accuracy from just word identity). This demonstrates that general primacy information is not sufficient to cause the grammatical representation of non-prototypical cases that we demonstrate in our previous results.

Conclusion BERT takes advantage of type-level information when it is available, in order to represent information about grammatical role. But, just as humans can understand sentences like “Man bites dog,” our probing task on non-prototypical subjects and objects reveals that, in higher layers of BERT, contextual information can override type-level biases using fine-grained syntactic word order information.

References

Louis Clouatre, Prasanna Parthasarathi, Amal Zouaq, and Sarath Chandar. 2021. Demystifying neural language models' insensitivity to word-order. *arXiv preprint arXiv:2107.13955*.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Richard Futrell, Evgeniia Diachek, Nafisa Syed, Edward Gibson, and Evelina Fedorenko. 2019. Formal marking is redundant with lexico-semantic cues to meaning in transitive clauses. Poster presented at the 32nd Annual CUNY Conference on Sentence Processing.

Edward Gibson, Steven T. Piantadosi, Kimberly Brink, Leon Bergen, Eunice Lim, and Rebecca Saxe. 2013. A noisy-channel account of crosslinguistic word-order variation. *Psychological Science*, 24(7):1079–1088.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. Bert & family eat word salad: Experiments with text understanding. *arXiv preprint arXiv:2101.03453*.

Jack Hessel and Alexandra Schofield. 2021. How effective is bert without word ordering? implications for language understanding and data privacy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 204–211.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Francis Mollica, Matthew Siegelman, Evgeniia Diachek, Steven T Piantadosi, Zachary Mineroff, Richard Futrell, Hope Kean, Peng Qian, and Evelina Fedorenko. 2020. Composition is the core driver of the language-selective network. *Neurobiology of Language*, 1(1):104–134.

Joe O'Connor and Jacob Andreas. 2021. What context features can transformer language models use? *arXiv preprint arXiv:2106.08367*.

Thang M Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? *arXiv preprint arXiv:2012.15180*.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *ACL*, pages 4609–4622.

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. *arXiv preprint arXiv:2104.06644*.

Can language models capture syntactic associations without surface cues? A case study of reflexive anaphor licensing in English control constructions

Soo-Hwan Lee
Department of Linguistics
New York University
soohwan.lee@nyu.edu

Sebastian Schuster
Department of Linguistics
Center for Data Science
New York University
schuster@nyu.edu

1 Introduction

Recent studies have shown that language models (LMs) have the ability to capture many long-distance dependencies such as filler-gap dependencies (Wilcox et al., 2018) and subject-verb agreement (Linzen et al., 2016) despite only learning from surface strings. However, this ability has primarily been shown for constructions for which the surface strings frequently provide information about dependencies in the form of agreement patterns. For example, if a model has access to sentences with and without a noun phrase intervening between the subject and the main verb (1), it is often able to infer the agreement dependencies from the surface string alone: (Linzen et al., 2016; Marvin and Linzen, 2018; Goldberg, 2019; Gulordava et al., 2018; Hu et al., 2020b). The surface cues are boldfaced in (1):

(1) The **girls** who the boy likes **are** smiling.

Importantly, such agreement patterns are not available for all constructions. Consider, for example, English control constructions with non-finite embedded clauses (2-3). The main verb in the embedded clause cannot be inflected and therefore the clause generally lacks agreement information. The main exception to this is when the embedded clause contains a reflexive anaphor (e.g., *himself*). In such cases, the anaphor refers to either the subject or the object in the higher clause (the *controller*) and thus has to agree with the controller. In (2), the anaphor *himself* is co-referential with the subject under the subject control predicate *promise*. In (3), the anaphor is co-referential with the object under the object control predicate *persuade*.

(2) The **artist** promised the lawyers to make fun of **himself**. (Subject control)

(3) The artists persuaded the **lawyer** to make fun of **himself**. (Object control)

Given the lack of agreement information on the verb, it is difficult to infer whether the controller should be the subject or the object of the matrix clause from the surface string alone, unless the embedded clause contains a reflexive anaphor. Such constructions, however, are almost non-existent in corpora.[1] Hence, LMs trained on naturalistic corpora likely fail to capture this type of dependency.

In this work, we examine a Transformer-based LM, namely Generative Pre-trained Transformer 2 (GPT-2) (Radford et al., 2019), which is trained only on surface strings, to see whether or not the model makes correct predictions about the agreement patterns of reflexive pronouns in subject and object control constructions. Our findings show that GPT-2 struggles with subject control constructions such as (2), but does quite well on object control constructions such as (3). One reason might be that the model tries to associate the anaphor with the closest noun phrase. Moreover, while we find that a model with a larger number of parameters shows higher accuracy on the tasks related to subject control constructions, performance remains below chance and the model does not mimic human behavior.

2 Language model

We evaluated to what extent an LM predicts the correct agreement patterns for subject and object control constructions involving a reflexive anaphor. Given its strong performance on many other syntactic tasks (Warstadt et al., 2020), we used GPT-2 (Radford et al., 2019) through the HuggingFace Transformer library (Wolf et al., 2020). GPT-2 uses a self-attention mechanism that enables it to learn to focus on certain parts of the input that are

[1] For example, the Corpus of Contemporary American English (Davies, 2008), which contains more than 1 billion words, includes exactly one example with *promise* in which a reflexive agrees with the controller.

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 206-211.
Held on-line February 7-9, 2022

		Condition	Example
With object	SUBJECT CONTROL		
	Promise	Baseline	The **lawyer** promised the artist to make fun of **himself**.
		Distractor	The **lawyer** promised the artists to make fun of **himself**.
		Ungrammatical	*The **lawyers** promised the artist to make fun of **himself**.
	Offer	Baseline	The **lawyer** offered the artist to make fun of **himself**.
		Distractor	The **lawyer** offered the artists to make fun of **himself**.
		Ungrammatical	*The **lawyers** offered the artist to make fun of **himself**.
	Guarantee	Baseline	The **lawyer** guaranteed the artist to make fun of **himself**.
		Distractor	The **lawyer** guaranteed the artists to make fun of **himself**.
		Ungrammatical	*The **lawyers** guaranteed the artist to make fun of **himself**.
	OBJECT CONTROL		
	Persuade	Baseline	The lawyer persuaded the **artist** to make fun of **himself**.
		Distractor	The lawyers persuaded the **artist** to make fun of **himself**.
		Ungrammatical	*The lawyer persuaded the **artists** to make fun of **himself**.
	Tell	Baseline	The lawyer told the **artist** to make fun of **himself**.
		Distractor	The lawyers told the **artist** to make fun of **himself**.
		Ungrammatical	*The lawyer told the **artists** to make fun of **himself**.
	Force	Baseline	The lawyer forced the **artist** to make fun of **himself**.
		Distractor	The lawyers forced the **artist** to make fun of **himself**.
		Ungrammatical	*The lawyer forced the **artists** to make fun of **himself**.
No object	SUBJECT CONTROL		
	Promise	Baseline	The **lawyer** promised to make fun of **himself**.
		Ungrammatical	*The **lawyers** promised to make fun of **himself**.
	Offer	Baseline	The **lawyer** offered to make fun of **himself**.
		Ungrammatical	*The **lawyers** offered to make fun of **himself**.
	Guarantee	Baseline	The **lawyer** guaranteed to make fun of **himself**.
		Ungrammatical	*The **lawyers** guaranteed to make fun of **himself**.

Table 1: Associates are **boldfaced**. Baseline, Distractor, Ungrammatical conditions are based on Hu et al. (2020a).

recognized to be more important for predicting the next word than others. The model is pre-trained on the WebText dataset (Radford et al., 2019) which is estimated to contain 8 billion tokens (see Warstadt et al., 2020). The corpus is tokenized into sub-word units using the byte pair encoding compression algorithm (Sennrich et al., 2016). GPT-2 is an autoregressive language model, that is, its pre-training objective is a next-token prediction task in which it aims to maximize the probability of each token given its previous tokens.

To examine whether an increase in the number of parameters affects performance on the agreement task, we evaluated two differently sized pre-trained GPT-2 models: GPT-2 (small) with ∼117 million parameters and GPT-2 XL with ∼1.5 billion parameters. Both models were trained on the same corpus and only differ in their number of parameters.

3 Experimental design

The frequency of each reflexive pronoun in English (e.g., *himself*, *herself*, and *themselves*) differs greatly from one another in many standard corpora (Hu et al., 2020a). In order to minimize this confound, we keep the reflexive word constant in all of our stimuli and vary the preceding context as little as possible. Table 1 shows our example stimuli with the reflexive anaphor, *himself*, embedded in a non-finite clause. We used *himself* instead of *herself*, since *himself* is usually more frequent than *herself* in corpora. We avoided using *themselves* mainly due to its number-neutral usage. Under our experimental design, the anaphor *himself* is associated with either the subject or the object in the matrix clause depending on the matrix predicate (e.g., *promise* or *persuade*). We used 5 noun phrases for subjects and objects, 3 matrix verbs for subject control, 3 matrix verbs for object control, and 5 embedded clauses (see Appendix A).

Adapting Hu et al.'s (2020a) experimental design, we generated grammatical sentences by matching the number of the reflexive anaphor and the controller (the *associates*) while being flexible about the number of the non-associate. The 'Baseline' condition consists of (non-)associates that always match in number. The 'Distractor' condition consists of a non-associate that differs from the associates in number. The associates are boldfaced and the non-associates are underlined in (4-5):

(4) The **lawyer** promised the artist to make fun of **himself**. (Baseline)

(5) The **lawyer** promised the artists to make fun of **himself**. (Distractor)

For the 'Ungrammatical' condition, the number of the associates are mismatched while the number of the anaphor and the non-associate are matched as shown in (6):

(6) *The **lawyers** promised the artist to make fun of **himself**. (Ungrammatical)

As mentioned in the previous section, GPT-2 assigns a probability to every token in a sentence based on its preceding tokens. For minimal pairs such as (4-6), we expect the probability assigned to *himself*, P(himself), in the 'Ungrammatical' condition to be lower than P(himself) in both the 'Baseline' and 'Distractor' conditions. Hence, chance accuracy is 33%. We constructed 100 minimal pairs for each of the matrix verbs shown in Table 1.

Since LM performance on reflexive anaphor licensing has generally been mixed (Marvin and Linzen, 2018; Futrell et al., 2019; Hu et al., 2020a), we also examined whether GPT-2 can make correct associations between the reflexive anaphor and the controller when there is no distracting noun (non-associate) intervening between the two. Hence, we examined simple control cases where the non-associate is absent using subject control constructions (7-8). Note that this is not possible with object control constructions, since neither the subject nor the object can be omitted.

(7) The **lawyer** promised to make fun of **himself**. (Baseline)

(8) *The **lawyers** promised to make fun of **himself**. (Ungrammatical)

We constructed 25 minimal pairs: 25 sentences for the 'Baseline' condition and 25 sentences for the 'Ungrammatical' condition. We expect P(himself) in the 'Ungrammatical' condition to be lower than P(himself) in the 'Baseline' condition. Hence, chance accuracy is 50%.

4 Results

Table 2 shows that GPT-2 (small)'s mean accuracy on subject control constructions with objects (4%) is significantly lower than its mean accuracy on object control constructions (100%). The larger GPT-2 XL shows higher accuracy on subject control constructions used with the matrix verbs *promise* (13% → 47%) and *offer* (0% → 20%). However, GPT-2 XL's accuracy on subject control constructions used with the matrix verb *guarantee* more or less remains the same (0% → 3%). The model's mean accuracy on subject control constructions with objects (23%) is thus still below chance accuracy (33%) and is significantly lower than its mean accuracy on object control constructions (97%). The results from the control experiment in Table 3 show that the poor performance on subject control with objects cannot be attributed to the issues related to reflexive anaphor licensing per se. Both models perform at ceiling on sentences without objects (100%), which suggests that the models are generally able to predict licensing patterns between reflexives and noun phrases based on number.

	GPT-2 (small)	GPT-2 XL
Promise	0.13	0.47
Offer	0.00	0.20
Guarantee	0.00	0.03
Mean	0.04	0.23
Persuade	1.00	0.95
Tell	1.00	0.95
Force	1.00	1.00
Mean	1.00	0.97

Table 2: GPT-2 performance on transitive subject and object control constructions (with object). Mean accuracy for each type of constructions is included. Chance accuracy is 0.33.

	GPT-2 (small)	GPT-2 XL
Promise	1.00	1.00
Offer	1.00	1.00
Guarantee	1.00	1.00
Mean	1.00	1.00

Table 3: GPT-2 performance on intransitive subject control constructions (no object). Mean accuracy is included. Chance accuracy is 0.50.

Taken together, the results suggest that both versions of GPT-2 primarily rely on the heuristic to associate the reflexive anaphor with the object NP. One likely reason for this behavior is that the reflexive anaphor is linearly closer to the object than to the subject. Given that syntactically complex sentences are not commonly represented in corpora (Marvin and Linzen, 2018), it is likely that the model learned to associate reflexives with the linearly closest noun phrase from naturalistic training corpora. Further, that both versions of GPT-2 perform similarly poorly suggests that an increase in the number of parameters does not lead to a considerable increase in accuracy.

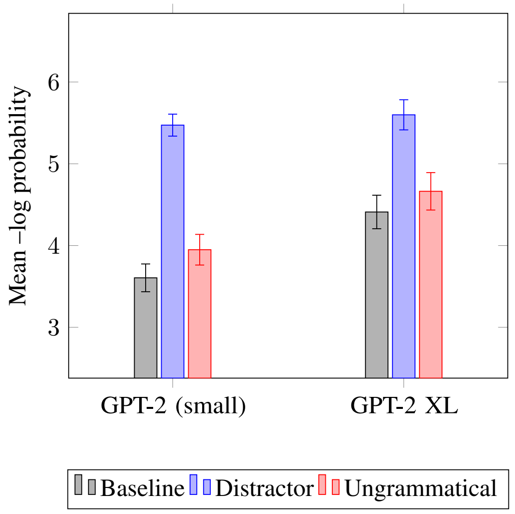


Figure 1: Mean negative log probability at the reflexive anaphor in transitive subject control constructions. Error bars indicate 95% confidence intervals.

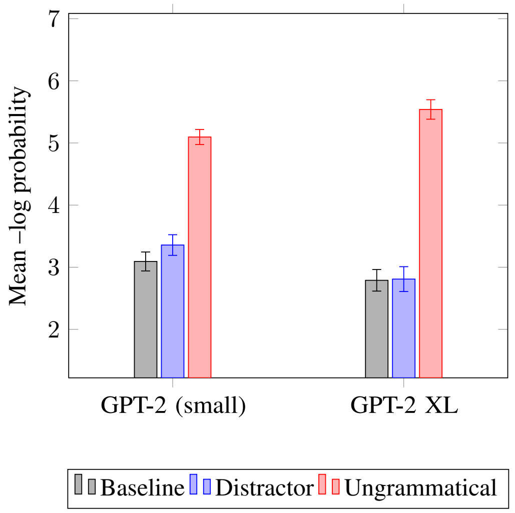


Figure 2: Mean negative log probability at the reflexive anaphor in transitive object control constructions. Error bars indicate 95% confidence intervals.

To further investigate the reason for the low performance on the agreement task for the transitive subject control constructions, we computed the mean surprisal values at the reflexive word *himself* for each of the 3 conditions. Figure 1 shows that, for the subject control constructions, both versions of GPT-2 have higher surprisal values in the 'Distractor' condition than in the 'Ungrammatical' condition, which provides additional evidence that the model adopts the strategy of agreeing with the closest NP. For object control constructions, on the other hand, both versions of GPT-2 show higher surprisal values in the 'Ungrammatical' condition than in the 'Distractor' condition (Figure 2), as already indicated by the near-perfect accuracy on the object control tasks. Moreover, we find that the surprisal of *himself* is almost identical in the conditions in which the object NP is singular ('Baseline' and 'Ungrammatical' for subject control constructions, and 'Baseline' and 'Distractor' for object control constructions), which further suggests that the model bases its predictions primarily on the number of the object NP in both types of constructions.

5 Discussion

The results from our experiments suggest that GPT-2 is unable to correctly distinguish subject control from object control constructions.[2] One potential strategy for increasing model accuracy is to augment the training data with examples of the form that we used for evaluation, which may lead models such as GPT-2 to learn the correct generalizations. However, while such a strategy may solve the problem for these specific constructions, the results that we presented here also highlight important limitations of training models from surface strings present in naturalistic corpora alone. This suggests that successfully mimicking human linguistic behavior may require a model that has access to meaning during training, as recently argued by Bender and Koller (2020), so that for example, it can learn the differences between subject and object control verbs (e.g., *promise* versus *persuade*).

Acknowledgements

We would like to thank Tal Linzen, Alec Marantz, and the anonymous reviewers for their thoughtful feedback. This material is based upon work supported by the National Science Foundation under Grant #2030859 to the Computing Research Association for the CIFellows Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the Computing Research Association.

[2] Some speakers of English also do not accept transitive subject control constructions (Courtenay, 1998). However, GPT-2 does not behave like this group of speakers either: If it did, it should assign similarly high surprisal values to all items with an object and a subject control verb, which is not what we observed in our experiments (see Figure 1).

References

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.

Karen Courtenay. 1998. Summary: Subject control verb PROMISE in English. https://linguistlist.org/issues/9/9-651/.

Mark Davies. 2008. The corpus of contemporary American English: 450 million words, 1990-present.

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, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint*, abs/1901.05287.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Jennifer Hu, Sherry Yong Chen, and Roger Levy. 2020a. A closer look at the performance of neural language models on reflexive anaphor licensing. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 323–333, New York, New York. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020b. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

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.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Stimuli

Noun phrases We manually constructed the following list of noun phrases: *the professor*, *the lawyer*, *the artist*, *the student*, and *the child*. The plural versions of the noun phrases were also used to generate grammatical and ungrammatical sentences. Each noun phrase is realized in the subject and object positions equally often in transitive sentences. Each noun phrase is realized with each of their matrix verbs equally often as well.

Matrix verbs The matrix verbs determine whether a given construction is subject or object control. For subject control verbs, we used *promise*, *offer*, and *guarantee*. For object control verbs, we used *persuade*, *tell*, and *force*.

Embedded clauses We manually constructed a list of non-finite embedded clauses hosting the reflexive anaphor *himself*: *to make fun of himself*, *to examine himself*, *to diagnose himself*, *to embarrass himself*, and *to disguise himself*. The embedded anaphor refers back to either the subject or the object depending on the matrix verb.

Incremental Acquisition of a Minimalist Grammar using an SMT-Solver

Sagar Indurkhya
MIT
32 Vassar St.
Cambridge, MA 02139
indurks@mit.edu

A central question in cognitive linguistics is how children everywhere can readily acquire *Knowledge of Language* (KoL) from (restrictive) Primary Linguistic Data (PLD) (Chomsky, 1986; Berwick et al., 2011; Chomsky, 2013). This study addresses this question by introducing a novel procedure, implemented as a working computer program, that uses an interactive theorem prover to incrementally infer a Minimalist Grammar (MG) (Stabler, 1996). The procedure, which is inspired by (Rayner et al., 1988) and builds on earlier work by (Indurkhya, 2020), takes the form of a computational model of child language acquisition (Berwick, 1985; Chomsky, 1965). The procedure takes as input a sequence of paired interface conditions - i.e. each entry is a Phonological Form (PF), encoding a sentence, paired with a Logical Form (LF), which encodes thematic roles for each predicate as well as agreement relations; the input sequence, which corresponds to the PLD that a child is exposed to, is organized into a sequence of batches that the procedure consumes incrementally. The procedure outputs an MG lexicon, consisting of a set of (word, feature-sequence) pairings, that yields, for each entry in the input sequence, a minimalist derivation that satisfies the listed interface conditions; the output MG lexicon corresponds to the KoL that the child acquires from processing the PLD.

The procedure, which models a child language learner, operates as follows. The initial state of the the learner is an empty MG lexicon. The procedure incrementally constructs an MG lexicon: at each step the procedure takes as input a batch of the PLD and the lexicon that constitutes the current state of the learner, and then it augments the input lexicon with the minimal set of additional lexical entries needed to ensure that the augmented lexicon will yield, for each entry in the batch of PLD, a minimalist derivation that satisfies the listed interface conditions.[1] When processing a batch of the PLD, the learner first constructs a set of logical formulae, expressed using the logic of Satisfiability Modulo Theories (SMT) (De Moura and Bjørner, 2011), that encodes: (i) an SMT-model of an MG lexicon that is required to have at least the lexical entries in the input lexicon; (ii) for each entry in the batch of PLD, an SMT-model of an MG derivation that must be derivable from the lexicon and that must satisfy the interface conditions listed for that entry.[2] The procedure then employs the Z3 SMT-solver (De Moura and Bjørner, 2008) to identify a solution to this set of SMT formulae that corresponds to the smallest[3] lexicon, and from this solution the "augmented" lexicon, which is the new state of the learner, is automatically recovered.[4] The final output of the procedure – i.e. the MG lexicon yielded after consuming the full PLD – corresponds to the KoL that the learner has acquired. Importantly, at a given step of the procedure, the size of the SMT-model of the lexicon is constrained by the size of the input lexicon (and a small, fixed, number of lexical entries that may be added), and the number of SMT-models of derivations is constrained by the size of the PLD-batch - *thus, the procedure can iteratively consume a large PLD without blowing up the size of the constructed SMT-models, thereby avoiding computational intractibility.*

[1] Each iteration of this process corresponds to an application of the instantaneous MG acquisition procedure introduced in (Indurkhya, 2020) and detailed in §3.2 of (Indurkhya, 2021a).

[2] The SMT-model of the lexicon is linked to each SMT-model of a derivation via common free-variables. See Ch. 2 of (Indurkhya, 2021a) for a complete presentations of these SMT-models.

[3] As measured by (firstly) the number of distinct feature sequences that appear in the lexicon, and (secondly) the total number of features that appear in the lexicon. Unlike (Indurkhya, 2020), here the acquisition procedure is restricted to work with a single selectional feature, x_0, which has the benefit of reducing the size of the SMT model, but yields a lexicon that is underconstrained w.r.t. c-selection; see (Indurkhya, 2021b) for a discussion of how model based collaborative filtering could be used to constrain which arguments a predicate can select within a derivation.

[4] This augmented lexicon is a superset of the input lexicon.

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 212-216.
Held on-line February 7-9, 2022

Batch	I_i	Interface	Interface Conditions
2	I_{29}	PF	john/N has asked/V whether pizza/N was eaten/V.
		LF	$\theta_{\text{asked}}[s\colon \text{john}, o\colon \text{whether pizza was eaten}]$, $Agr_{\text{has}}[s\colon \text{john}]$, $\theta_{\text{eaten}}[o\colon \text{pizza}]$, $Agr_{\text{was}}[s\colon \text{pizza}]$
	I_{30}	PF	mary/N was told/V that john/N has eaten/V pizza/N.
		LF	$\theta_{\text{told}}[o\colon \text{that john has eaten pizza}, i\colon \text{mary}]$, $Agr_{\text{was}}[s\colon \text{mary}]$, $\theta_{\text{eaten}}[s\colon \text{john}, o\colon \text{pizza}]$, $Agr_{\text{has}}[s\colon \text{john}]$
	I_{31}	PF	mary/N has told/V john/N that icecream/N was eaten/V.
		LF	$\theta_{\text{told}}[s\colon \text{mary}, o\colon \text{that icecream was eaten}, i\colon \text{john}]$, $Agr_{\text{has}}[s\colon \text{mary}]$, $\theta_{\text{eaten}}[o\colon \text{icecream}]$, $Agr_{\text{was}}[s\colon \text{icecream}]$
	I_{32}	PF	mary/N has asked/V john/N whether she/N was eating/V pizza/N.
		LF	$\theta_{\text{asked}}[s\colon \text{mary}, o\colon \text{whether she was eating pizza}, i\colon \text{john}]$, $Agr_{\text{has}}[s\colon \text{mary}]$, $\theta_{\text{eating}}[s\colon \text{she}, o\colon \text{pizza}]$, $Agr_{\text{was}}[s\colon \text{she}]$
	I_{33}	PF	who has mary/N told/V that she/N was eating/V icecream/N?
		LF	$\theta_{\text{told}}[s\colon \text{mary}, o\colon \text{that she was eating icecream}, i\colon \text{who}]$, $Agr_{\text{has}}[s\colon \text{mary}]$, $\theta_{\text{eating}}[s\colon \text{she}, o\colon \text{icecream}]$, $Agr_{\text{was}}[s\colon \text{she}]$
	I_{34}	PF	who was asked/V whether mary/N has given/V john/N money/N?
		LF	$\theta_{\text{asked}}[o\colon \text{whether mary has given john money}, i\colon \text{who}]$, $Agr_{\text{was}}[s\colon \text{who}]$, $\theta_{\text{given}}[s\colon \text{mary}, o\colon \text{money}, i\colon \text{john}]$, $Agr_{\text{has}}[s\colon \text{mary}]$
3	I_{35}	PF	who has told/V john/N everything/N that mary/N was asked/V?
		LF	$\theta_{\text{told}}[s\colon \text{who}, o\colon \text{everything that mary was asked}, i\colon \text{john}]$, $Agr_{\text{has}}[s\colon \text{who}]$, $\theta_{\text{asked}}[o\colon \text{everything}, i\colon \text{mary}]$, $Agr_{\text{was}}[s\colon \text{mary}]$
	I_{36}	PF	was someone/N given/V everything/N that she/N has eaten/V?
		LF	$\theta_{\text{given}}[o\colon \text{everything that she has eaten}, i\colon \text{someone}]$, $Agr_{\text{was}}[s\colon \text{someone}]$, $\theta_{\text{eaten}}[s\colon \text{she}, o\colon \text{everything}]$, $Agr_{\text{has}}[s\colon \text{she}]$
4	I_{37}	PF	mary/N has seen/V everyone/N who john/N was eating/V.
		LF	$\theta_{\text{seen}}[s\colon \text{mary}, o\colon \text{everyone who john was eating}]$, $Agr_{\text{has}}[s\colon \text{mary}]$, $\theta_{\text{eating}}[s\colon \text{john}, o\colon \text{everyone}]$, $Agr_{\text{was}}[s\colon \text{john}]$
	I_{38}	PF	john/N has seen/V someone/N who was eating/V icecream/N.
		LF	$\theta_{\text{seen}}[s\colon \text{john}, o\colon \text{someone who was eating icecream}]$, $Agr_{\text{has}}[s\colon \text{john}]$, $\theta_{\text{eating}}[s\colon \text{someone}, o\colon \text{icecream}]$, $Agr_{\text{was}}[s\colon \text{someone}]$
	I_{39}	PF	john/N has seen/V someone/N who was eaten/V.
		LF	$\theta_{\text{seen}}[s\colon \text{john}, o\colon \text{someone who was eaten}]$, $Agr_{\text{has}}[s\colon \text{john}]$, $\theta_{\text{eaten}}[o\colon \text{someone}]$, $Agr_{\text{was}}[s\colon \text{someone}]$

Figure 1: The acquisition procedure incrementally inferred the grammar listed in Fig. 2 after (successively) consuming four batches of primary linguistic data (PLD), the latter three of which are shown here (see Table 3.2 in (Indurkhya, 2021a) for the first batch). Each of the input sentences listed here has one degree (level) of embedding. The second batch consist of sentences in which the embedded clause is a declarative (e.g. I_{31}) or an interrogative (e.g. I_{34}). The third and fourth batch consists of sentences with an embedded (restrictive) relative clause. Notably, in the case of LF interface conditions that mark an embedded clause as an argument, the tokens making up the embedded clause are interpreted as a multi-set of phonological forms - e.g. the LF interface conditions for I_{35} indicate that the phrase to be formed from the multi-set of phonological forms $\{$ everything, that, mary, was, asked $\}$ will serve as an internal argument of the (lexical) verb "told". Hence, the LF interface conditions *do not explicitly encode any information about the linear ordering of the phonological forms that form the sentence, and only serve to constrain the hierarchical relations that establish predicate-argument structure.*

ID	Category	Features	Phonological Forms																															
			what	who	***that***	***whether***	has	was	***she***	everything	someone	icecream	money	mary	pizza	john	nothing	boy	story	given	told	asked	known	***seen***	asking	eating	eaten	to	***everyone***	sleeping	slept	the	a	ϵ
$\mathfrak{L}_1$	V	$= x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	×	×	×	×	·	·	·	·	·	·	·
$\mathfrak{L}_2$	V	$= x_0, = x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_3$	$C_{decl.}$	$= x_0, C$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_4$	$C_{ques.}$	$<= x_0, +z, C$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_5$	v	$<= x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_6$	$C_{ques.}$	$<= x_0, C$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_7$	v	$<= x_0, = x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×
$\mathfrak{L}_8$	P	$= x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	·	·	·	·	·	·
$\mathfrak{L}_9$	D	$= x_0, \sim x_0, -l$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	·
$\mathfrak{L}_{10}$	D	$= x_0, \sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	·
$\mathfrak{L}_{11}$	D	$\sim x_0, -z$	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{12}$	D	$\sim x_0, -l, -z$	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{13}$	T	$= x_0, +l, \sim x_0$	·	·	·	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{14}$	V	$\sim x_0$	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	×	·	·	·
$\mathfrak{L}_{15}$	N	$\sim x_0, -l$	·	·	·	·	·	·	×	×	×	×	×	×	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{16}$	N	$\sim x_0$	·	·	·	·	·	·	·	×	×	×	×	×	×	×	×	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{17}$	$C_{decl.}$	$= x_0, \sim x_0$	·	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{18}$	$C_{decl.}$	$= x_0, +z, \sim x_0$	·	×	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
$\mathfrak{L}_{19}$	N	$\sim x_0, -z$	·	·	·	·	·	·	·	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	×	·	·	·	·	·
$\mathfrak{L}_{20}$	N	$\sim x_0, -l, -z$	·	·	·	·	·	·	·	·	×	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·

Figure 2: A factored representation of an MG lexicon inferred by the acquisition procedure - an X indicates that a lexical entry pairing a feature sequence with phonological form. Bold horizontal-lines dividing the feature-sequences indicate which of the four (successive) PLD batches was processed when those feature-sequences were added; notably, the lexicon was augmented with only four feature-sequences ($\mathfrak{L}_{17}$ - $\mathfrak{L}_{20}$) to handle the embedded clauses found in PLD batches 2-4. Bolded phonological forms appeared after the first PLD batch - e.g. "she" first appears in the second PLD batch, and is paired with $\mathfrak{L}_{15}$ (that was added upon processing the first PLD batch).

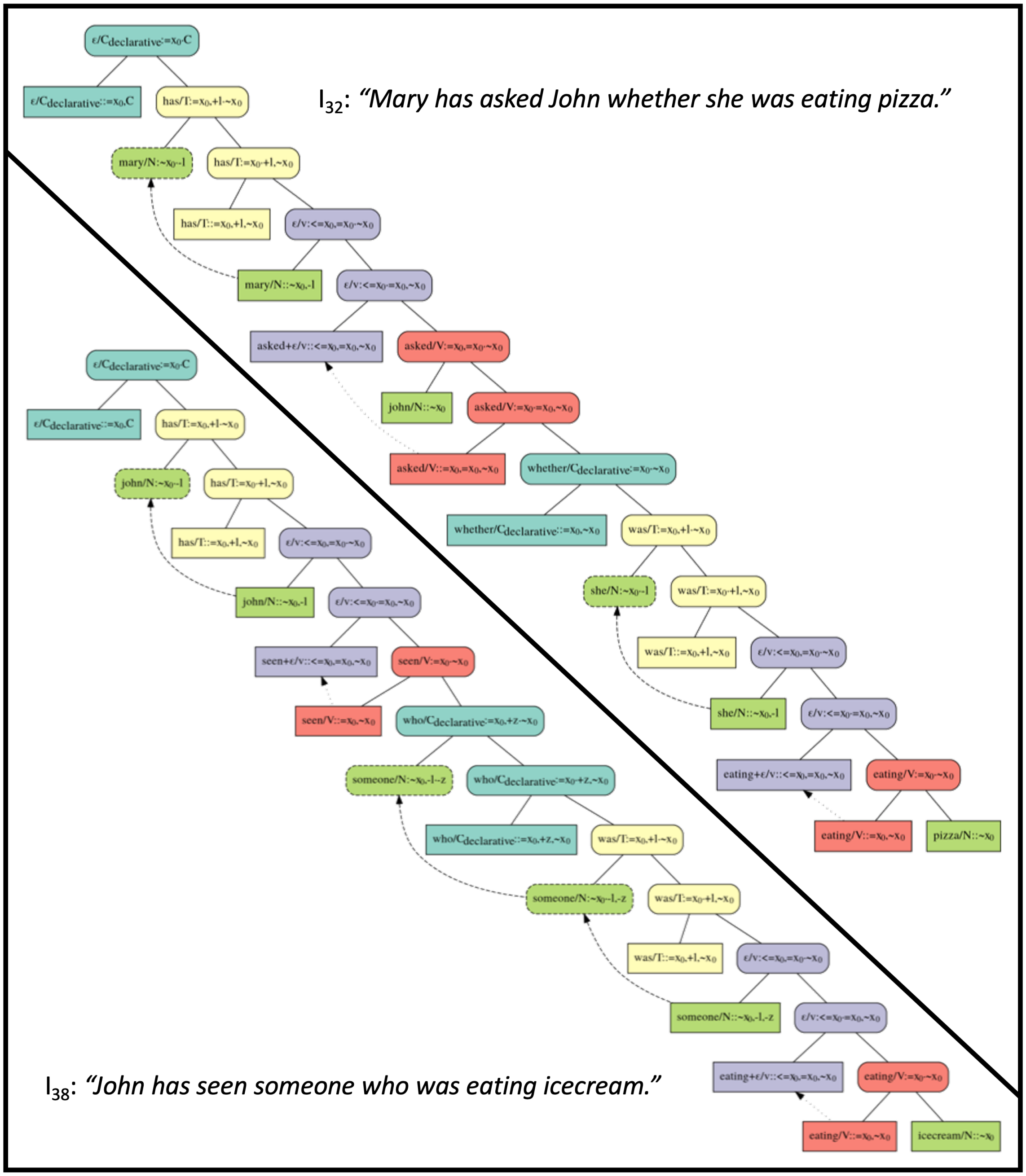


Figure 3: Presentations of two MG derivations identified by the acquisition procedure that satisfy the interface conditions listed in I_{32} and I_{38} respectively and that may be yielded by the lexicon listed in Fig. 2. The derivation for I_{32}, which derives a sentence with an embedded question, employs feature-sequences $\{\mathfrak{L}_1, \mathfrak{L}_2, \mathfrak{L}_3, \mathfrak{L}_7, \mathfrak{L}_{13}, \mathfrak{L}_{15}, \mathfrak{L}_{16}, \mathfrak{L}_{17}\}$, and the derivation for I_{38}, which derives a sentence with an embedded restrictive relative clause, employs feature-sequences $\{\mathfrak{L}_1, \mathfrak{L}_3, \mathfrak{L}_7, \mathfrak{L}_{13}, \mathfrak{L}_{15}, \mathfrak{L}_{16}, \mathfrak{L}_{18}, \mathfrak{L}_{20}\}$. The leaf nodes (indicated by absence of rounded corners) are lexical items selected from the lexicon. The derivation is assembled in a bottom-up manner via repeated applications of the structure-building operation *merge*. The feature sequences displayed in non-leaf nodes (indicated by rounded corners) have a dot, $\cdot$, that separates those features that have already been consumed (on the left) from those that have not (on the right) - see (Stabler, 2001) for details the MG feature system. Nodes with the same head have the same color. Head-movement is indicated by the dotted-arrows, and phrasal movement is indicated by the dashed arrows. Note that as the system is only provided with two types of complementizers, $C_{decl.}$ and $C_{ques.}$, the system re-uses the declarative complementizer sub-category for both types of embedded clauses; expanding the set of (sub-)categories available to the system is one possible avenue of future improvement to the system.

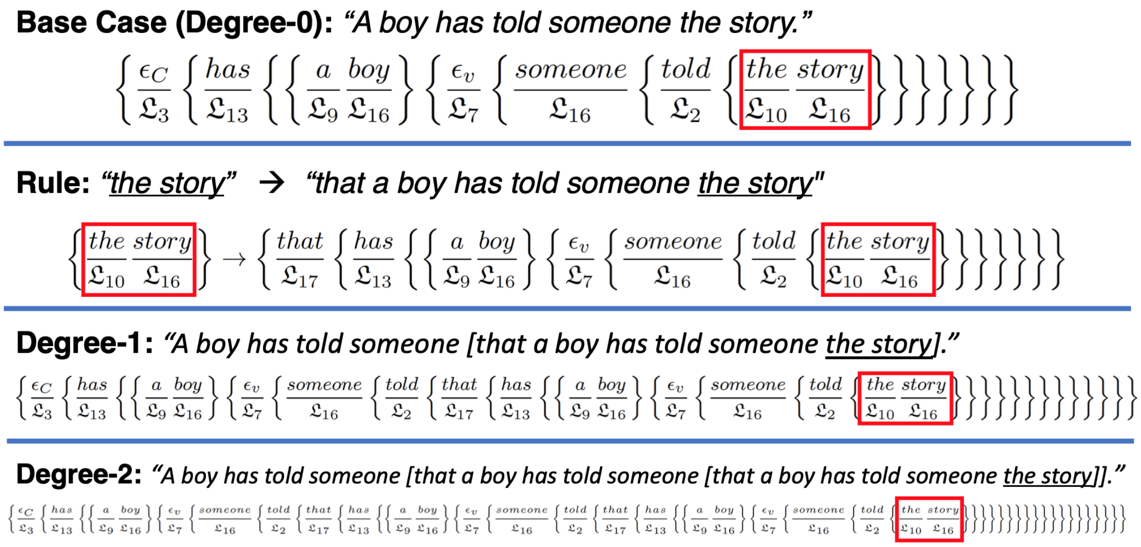

Base Case (Degree-0): "A boy has told someone the story."
Rule: "the story" → "that a boy has told someone the story"
Degree-1: "A boy has told someone [that a boy has told someone the story]."
Degree-2: "A boy has told someone [that a boy has told someone [that a boy has told someone the story]]."

Figure 4: A demonstration of how, by repeated application of a substitution-rule to a base-case derivation, the lexicon (listed in Fig. 2) that the acquisition procedured inferred from a PLD restricted to sentences with degree-0/1 embedding can yield a sentence with degree-n embedding for any $n \geq 0$. Note that these derivations only show what external merge operations would be applied, with internal merge assumed to immediately and automatically be applied whenever possible in the course of a derivation.

We demonstrate the capabilities of the acquisition procedure by using it to infer an MG lexicon from a PLD consisting of 39 simple sentences that were divided into four consecutive batches (having 28, 6, 2 and 3 entries respectively), with the first batch having sentences without any embedding, and the remaining batches (presented in Table 1) consisting of sentences with at most one degree of embedding (i.e. embedded declaratives or relative clauses). The procedure outputs an MG lexicon (see Fig. 2) that yields derivations for declaratives, yes/no-questions, and wh-questions in both active and passive voice; these derivations involve various forms of syntactic movement including wh-raising, subject-raising, T-to-C head-movement and V-to-v head-movement; the lexicon also includes entries for covert complementizers and light-verbs. The inferred lexicon aligns with contemporary theories of minimalist syntax[5] in so far as: (i) the lexicon yields the prescribed derivations for a variety of syntactic structures, utilizing syntactic movement (including head-movement) and covert lexical items as needed (see Fig. 3 for examples); (ii) expressions with related interpretations are assigned derivations systematically related by structural transformations. Furthermore, this lexicon can generate a countably infinite set of minimalist derivations, including derivations with n-levels of embedding for any $n \geq 0$, thereby generalizing beyond the input PLD (see Fig. 4 for more details).[6] Notably, the procedure does this without being provided a treebank of minimalist derivations that serve as examples of what the acquired lexicon should be able to yield, and to that end, the procedure constitutes a novel scheme for unsupervised inference of MGs.

The acquisition procedure demonstrates how an SMT-solver can aid in the study of linguistic theory: the solver enables us to separate out the questions of what KoL the learner acquires and how the learner acquires it – i.e. we can setup computational experiments in which we focus on specifying the learner's initial state and the conditions that the learner's final state must satisfy (w.r.t. the PLD), and leave to the solver questions of how the language-acquisition device goes from the initial state to the final state and what that final state is.

Acknowledgements

I would like to thank three anonymous reviewers as well as Robert C. Berwick, Sandiway Fong and Sanjoy Mitter for their comments and feedback.

[5] As presented in (Adger, 2003; Hornstein et al., 2005; Radford, 2009; Collins and Stabler, 2016).

[6] Cf. Neural-network based UD parsing frameworks that have difficulty generalizing from degree-0/1 embedding sentences to correctly parse degree-n embedding sentences for $n \geq 2$. See (Indurkhya et al., 2021) for details.

References

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

Robert C. Berwick. 1985. *The acquisition of syntactic knowledge*. MIT Press.

Robert C. Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive Science*, 35(7):1207–1242.

Noam Chomsky. 1965. *Aspects of the theory of syntax*. MIT Press.

Noam Chomsky. 1986. *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group.

Noam Chomsky. 2013. Poverty of the stimulus: Willingness to be puzzled. In *Rich languages from poor inputs*, pages 61–67. Oxford University Press.

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

Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient smt solver. In *Proceedings of the Theory and Practice of Software*, TACAS'08/ETAPS'08, pages 337–340, Berlin, Heidelberg. Springer-Verlag.

Leonardo De Moura and Nikolaj Bjørner. 2011. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77.

Norbert Hornstein, Jairo Nunes, and Kleanthes K Grohmann. 2005. *Understanding minimalism*. Cambridge University Press.

Sagar Indurkhya. 2020. Inferring minimalist grammars with an SMT-solver. *Proceedings of the Society for Computation in Linguistics*, 3(1):476–479.

Sagar Indurkhya. 2021a. *Solving for syntax*. Ph.D. thesis, Massachusetts Institute of Technology.

Sagar Indurkhya. 2021b. Using collaborative filtering to model argument selection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 629–639, Held Online. INCOMA Ltd.

Sagar Indurkhya, Beracah Yankama, and Robert C. Berwick. 2021. Evaluating Universal Dependency parser recovery of predicate argument structure via CompChain analysis. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 116–128, Online. Association for Computational Linguistics.

Andrew Radford. 2009. *An introduction to English sentence structure*. Cambridge University Press.

Manny Rayner, Asa Hugosson, and Goran Hagert. 1988. Using a logic grammar to learn a lexicon. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

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

Edward P Stabler. 2001. Recognizing head movement. In *International Conference on Logical Aspects of Computational Linguistics*, pages 245–260. Springer.

The interaction between cognitive ease and informativeness shapes the lexicons of natural languages

Thomas Brochhagen[*] **Gemma Boleda**[*] [†]

[*]Universitat Pompeu Fabra
[†]ICREA
{firstname.surname}@upf.edu

Introduction

Across languages, it is common for words to be associated with multiple meanings. Moreover, certain meanings are expressed by the same form more often than others (Jackson et al., 2019; Xu et al., 2020). For instance, the colexification –i.e., the conventional association of multiple meanings with the same form– of TOE and FINGER is found in at least 135 languages (Rzymski et al., 2020). These languages are spoken throughout the world and span multiple unrelated language families.

Recent research suggests that semantic relatedness increases colexification likelihood (Xu et al., 2020). Semantic memory may favor colexifying meanings that are easy to relate to one another. This, in turn, may aid vocabulary acquisition, lexical retrieval and interpretation. Building on these findings, we investigate the interplay between this and another major force: pressure for the lexicon to be informative, in the sense of supporting accurate information transfer (e.g., Regier et al., 2015). We hypothesize that languages strike a balance between these two forces. In particular, we expect colexification likelihood to increase with semantic relatedness, until a point is reached at which meanings are too related; for these highly related meanings, we expect pressure for informativeness to counteract the increasing trend, because these meanings would not be easy to disambiguate even in context. We find support for this hypothesis in two large scale analyses.[1]

Analysis 1

To study the relationship between semantic relatedness and colexification, we fit three generalized additive logistic models to colexification data spanning over 1200 languages and more than 1400 meanings, totaling 203056 data points. This data comes from CLICS^3 (Rzymski et al., 2020), the largest cross-linguistic database of colexifications available to date. The models characterize how likely a pair of meanings is to colexify in a given language as a function of one of three data-induced estimates of relatedness: distributional similarity, using pre-trained embeddings (Grave et al., 2018); associativity data (De Deyne et al., 2018); and the first principal component of these two measures (PC1). Both distributional and associative information are based on Dutch and English glosses of the meanings found in CLICS^3; that is, Dutch and English words are used as surrogates for meanings to estimate the latter's relatedness. Since language contact and common linguistic ancestry influence colexification (Jackson et al., 2019; Xu et al., 2020), the models are also passed information about how often a pair of meanings colexifies in other languages. This information is weighted by the phylogenetic/geographic distance to the response language. An indicator codifies whether a relatedness estimate stems from Dutch or English data.

Model comparison using approximate leave-one-out cross-validation suggests that PC1 is the best predictor of colexification, with a difference of -715 in expected log pointwise predictive density to the second highest ranked model. Figure 1 shows its estimated marginal effects. These results largely support to our hypothesis: colexification increases with relatedness until meanings are "too related", which makes their colexification decrease. Note, however, that the data are also consistent with a plateau rather than a decrease for highly related meanings (see shaded area in the figure). This is still consistent with the main hypothesis –informativeness counteracting simplicity for highly related meanings–, with a smaller effect of informativeness than we had expected.

[1] The manuscript that this abstract is based on is found at https://psyarxiv.com/efs4p

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 217-219.
Held on-line February 7-9, 2022

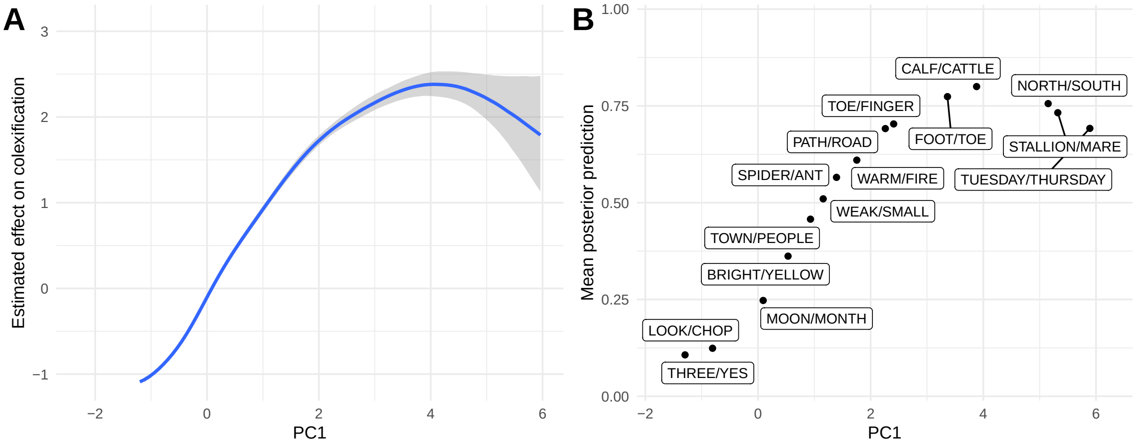


Figure 1: A: Marginal effects of standardized PC1. Shading shows 95% credible intervals. The smooth function $s(\cdot)$ characterizes how PC1's contribution to colexification likelihood changes across values. B: Mean posterior predictions for exemplary meaning pairs across PC1 values.

Analysis 2

Our hypothesis specifically predicts that the decrease in colexification likelihood for highly related meanings is due to their confusability. We next probe confusability more directly, focusing on the kind of relationship meanings stand in.

Pressure for informativeness should make colexifying opposites (e.g., LEFT and RIGHT) less likely than colexifying meanings in other kinds of relationships. Opposite meanings express contrasts, being maximally similar in every respect but one (e.g., Kliegr and Zamazal, 2018). Therefore, losing the distinction they encode can be expected to be particularly harmful in communicative terms. We compare opposites to meaning pairs standing in two semantic relations that do not necessarily lead to high confusability: part-whole (e.g., TOE-FOOT) and subsumption (e.g., CALF-CATTLE).

Colexification rates were estimated from 1416 meanings and 2279 languages from CLICS^3. Semantic relations are from WordNet (Fellbaum, 2015), using English words as proxies for meanings. Pairs in none of the three relations were classified as 'none/other'. As expected, this group has the lowest mean percentage of colexification (0.06, with a 95% CI of $[0.06, 0.06]$), followed by opposites $(1.4\ [1.3, 1.5])$, then by subsumption $(3.1\ [3.0, 3.3])$ and part-whole pairs $(3.7\ [3.5, 3.8])$. These results suggest, first, that standing in one of the three relations increases the odds for meanings to colexify compared to 'none/other'; and second, that not all relations are equally conducive to colexification, with opposites being less likely to colexify. We thus again find that relatedness makes colexification more likely, but that the need to distinguish confusable meanings can counteract this trend. Under our interpretation, simplicity makes colexification likelihood for opposites increase, whereas informativeness makes them decrease, resulting in their position in the middle compared to the other relations.

Conclusions

A growing body of research supports the idea that languages are efficient in the sense that they strike a good balance between informativeness and simplicity (e.g., Christiansen and Chater, 2008; Regier et al., 2015). Our large scale analyses suggest such a balance in the lexicon. We find that colexification likelihood increases with semantic relatedness, until an inflection point is reached, after which it decreases or flattens out (Analysis 1). This shift may be a consequence of a need for meanings to be distinguishable in context (Analysis 2).

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 715154). This paper reflects the authors' view only, and the EU is not responsible for any use that may be made of the information it contains.

References

Morten H. Christiansen and Nick Chater. 2008. Language as shaped by the brain. *BBS*, 31(05).

Simon De Deyne, Danielle J. Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. 2018. The "Small World of Words" English word association norms for over 12,000 cue words. *BRM*, 51(3):987–1006.

Christiane Fellbaum. 2015. *WordNet*. OUP.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proc. LREC*.

Joshua C. Jackson, Joseph Watts, Teague R. Henry, Johann-Mattis List, Robert Forkel, Peter J. Mucha, Simon J. Greenhill, Russell D. Gray, and Kristen A. Lindquist. 2019. Emotion semantics show both cultural variation and universal structure. *Science*.

Tomáš Kliegr and Ondřej Zamazal. 2018. Antonyms are similar: Towards paradigmatic association approach to rating similarity in SimLex-999 and WordSim-353. *DKE*, 115:174–193.

Terry Regier, Charles Kemp, and Paul Kay. 2015. *Word Meanings across Languages Support Efficient Communication*, chapter 11. John Wiley & Sons, Ltd.

Christoph Rzymski, Tiago Tresoldi, Simon J Greenhill, Mei-Shin Wu, et al. 2020. The database of cross-linguistic colexifications, reproducible analysis of cross-linguistic polysemies. *Sci. Data*, 7(1):1–12.

Yang Xu, Khang Duong, Barbara C Malt, Serena Jiang, and Mahesh Srinivasan. 2020. Conceptual relations predict colexification across languages. *Cognition*.

Learning constraints on *wh*-dependencies by learning how to efficiently represent *wh*-dependencies: A developmental modeling investigation with Fragment Grammars

Niels Dickson **Lisa Pearl** **Richard Futrell**
Department of Language Science
University of California, Irvine
{nielsd, lpearl, rfutrell}@uci.edu

1 Introduction

It's hotly contested how children learn constraints on the allowed forms in their language, such as constraints on *wh*-dependencies (these constraints are sometimes called *syntactic islands*: Chomsky 1973; Pearl and Sprouse 2013). When learning this knowledge, a prerequisite is knowing how to represent *wh*-dependencies – constraints can then be hypothesized over these dependency representations. Previous work (Pearl and Sprouse, 2013; Liu et al., 2019) explained disparate sets of syntactic island constraints by assuming different *wh*-dependency representations, without a unified dependency representation capturing all these constraints. Here, we implement a modeled learner attempting to learn a Fragment Grammar (**FG**) representation (O'Donnell et al., 2011; O'Donnell, 2015) of *wh*-dependencies—a representation comprised of potentially different-sized fragments that combine to form full dependencies—that best accounts for the input while being as compact as possible. In particular, FG implements a theory of efficiency that balances the size of the fragments in the resulting grammar while also maximizing the probability of the dependency structures comprised of these fragments. So, when deciding on the fragments to represent from linguistic input, a learner can choose between smaller fragments of the input that may be reused often in different contexts and larger fragments that can be accessed without building up the structure from smaller pieces. The resulting fragment-based *wh*-dependency representation can then be used to generate any *wh*-dependency's probability on the basis of its fragments, and so predict acceptability patterns for stimuli sets that reveal syntactic island knowledge. We find that the identified FG, learned from a realistic sample of *wh*-dependencies from English-learning children's input, can generate the attested acceptability judgment patterns for all syntactic islands previously investigated, highlighting how implicit knowledge of *wh*-dependency constraints can emerge from trying to learn to efficiently represent *wh*-dependencies more generally. We additionally compare the FG representation's performance against baselines inspired by previous proposals, finding that one baseline also yields equivalent performance. We discuss how this baseline is similar to and different from the FG representation.

2 *Wh*-dependency representation

We assume *wh*-dependencies are represented as sequences of phrase structure nodes that indicate the path from the gap to the *wh*-word (Pearl and Sprouse, 2013) (1a)-(1b). However, it's unknown whether the phrasal categories (e.g., CP, VP) in this representation need to be lexically subcategorized. For instance, does the dependency path for a *wh*-dependency with *claim* need to include that the verb is *claim* (1d) or not (1e)?

(1) What did Lily claim that Jack forgot?

a. What did [$_{IP}$ Lily [$_{VP}$ claim [$_{CP}$ that [$_{IP}$ Jack [$_{VP}$ forgot $\_\_{}_{what}$]]]]]?

b. phrase-structure nodes in syntactic path: IP-VP-CP-IP-VP

c. lexical information for those nodes: IP=*past*, VP=*claim*, CP=*that*, IP=*past*, VP=*forget*

d. possible representations with lexically-subcategorized VP *claim*: IP-VP$_{claim}$-CP-IP-VP, IP$_{past}$-VP$_{claim}$-CP-IP$_{past}$-VP, IP-VP$_{claim}$-CP$_{that}$-IP-VP$_{forget}$, ...

e. possible representations without lexically-subcategorized VP *claim*: IP-VP-CP-IP-VP, IP-VP-CP$_{that}$-IP-VP, IP$_{past}$-VP-CP$_{that}$-IP$_{past}$-VP$_{forget}$, ...

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 220-224.
Held on-line February 7-9, 2022

Figure 1 illustrates the consequences of lexical subcategorization and the related balance of fragment size mentioned earlier, showing two extremes of how to represent the example dependency path from (1). The leftmost representation uses minimal-sized fragments (phrasal-only like IP-VP, and lexicalized like IP$_{past}$) that may be reused often because they can appear in many different dependencies. This representation has no lexical subcategorization because the lexical information is separate from the phrasal structure. The rightmost representation uses a maximal-sized fragment (representing the entire dependency with both its phrasal structure and lexical pieces) that will only be reused if this exact dependency occurs. This representation has complete lexical subcategorization because all the lexical information is included in this phrase structure fragment. In terms of maximizing the probability of the dependency, each extreme has its drawbacks: the representation relying on minimal-sized fragments requires combining many individual fragments, which can lead to a lower probability even if the individual fragments have higher probabilities; the representation relying on the maximal-sized fragment likely has a fairly low probability unless this particular dependency happens to occur very frequently (and even if it does, this won't be true for all dependencies). To maximize the probability of a dependency in general, a better approach is to find some intermediate representation, such as the middle one in Figure 1, that involves some larger phrasal fragments incorporating lexical subcategorization (e.g., IP$_{past}$-VP), as well as some lexical-only fragments (e.g., VP$_{forget}$). In this example intermediate representation, there is thus a tradeoff between larger fragments that don't have to be built every time from smaller fragments (e.g., IP$_{past}$-VP from IP$_{past}$ and IP-VP) and smaller, more frequently-reused fragments (e.g., VP$_{forget}$). Of course, there are many possible intermediate representations, and the goal for a learner is to identify the best one that maximizes this tradeoff and so yields high probabilities collectively for the dependencies in the input.

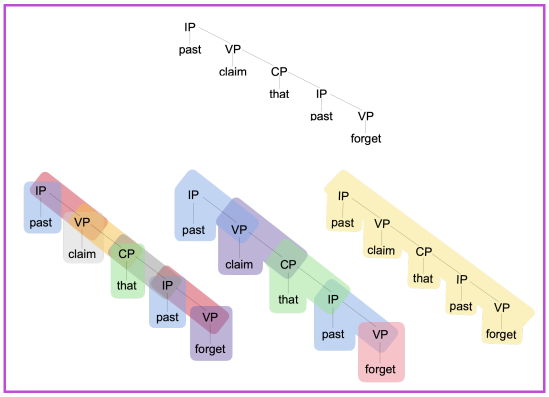


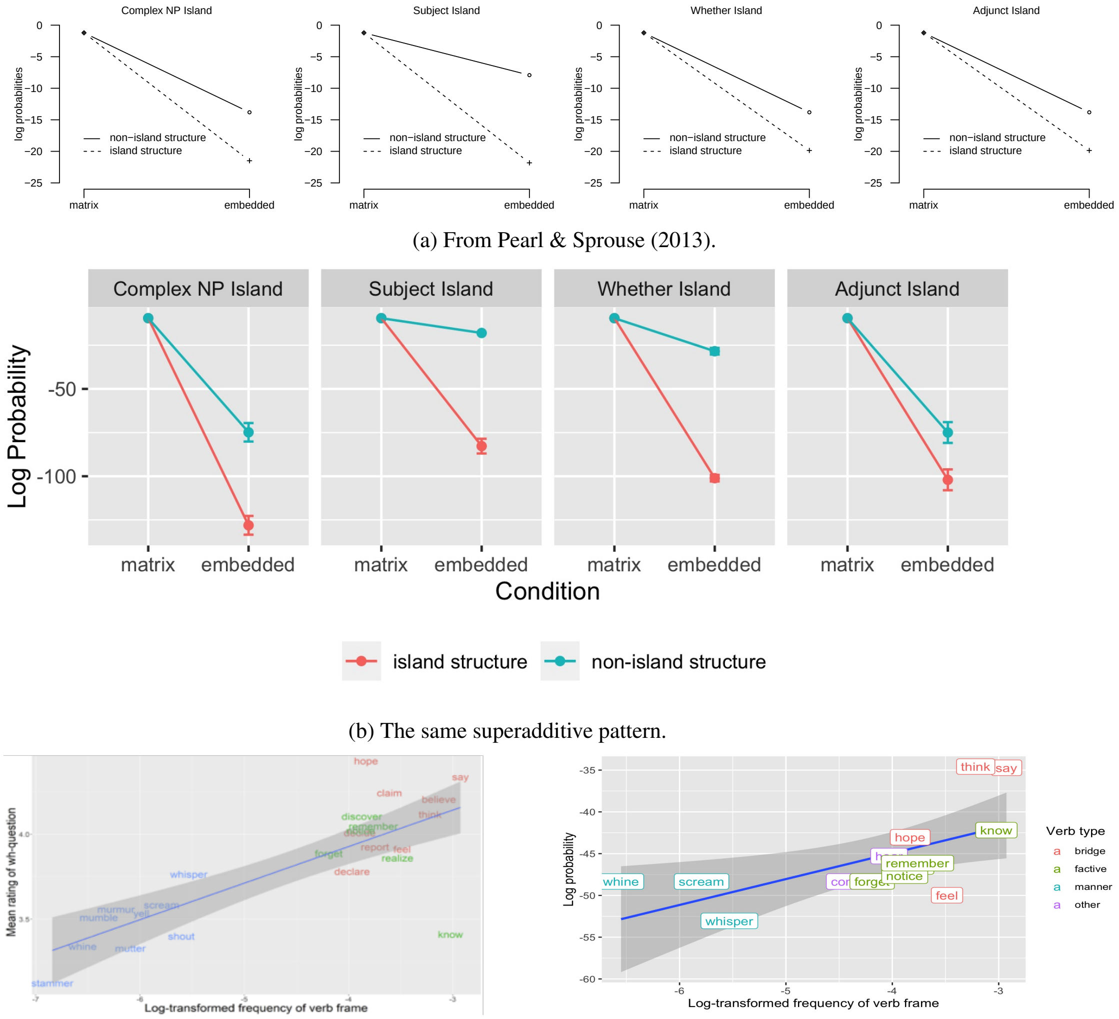
Figure 1: Example *wh*-dependency path as a syntactic tree and possible ways to build it from fragments.

3 Previous representation proposals

Previous developmental modeling work by Pearl and Sprouse (2013) predicted attested adult judgment patterns for 4 islands (Complex NP, Subject, Adjunct, Whether)—see Figure 2a—by assuming only CPs were lexically subcategorized (i.e., only the lexical information of CPs was included with the phrasal structure). Previous empirical work by Liu et al. (2019) predicted attested judgment patterns for 14 bridge (e.g, *say*), factive (e.g., *know*), and manner-of-speaking (e.g., *whisper*) verbs—see Figure 2c—in terms of the lexical frequency of the main-clause VP. While Liu et al. didn't explicitly propose a theory of representation, their results are compatible with a representation that lexically subcategorizes main-clause VPs (i.e., only the lexical information of the main verb is included with the phrasal structure). Yet, these are only two of many possible types of hypotheses for how the phrasal structure of *wh*-dependencies could be represented (i.e., different intermediate representations). Using an FG, we can explore the entire hypothesis space that investigates not only which lexical information should be included (e.g., CPs or main VPs), but also what size fragments are the most efficient for the phrasal structure of the dependency to be built from. Importantly, instead of telling the learner beforehand what phrase structure nodes are lexicalized and what size fragments to use, the learner using FGs infers both on the basis of its input.

4 Learning efficient representations that underlie *wh*-dependency constraints

We implement a computational-level modeled learner that attempts to identify an FG encoding the most efficient dependency path representation. The model uses Bayesian inference to identify the best representation. In particular, the modeled learner uses a Metropolis-Hastings-based inference algorithm to find the set of fragments that best explains the input, by yielding a high probability for the dependencies in the input. To identify this FG representation, the modeled learner uses the Metropolis-



(a) From Pearl & Sprouse (2013).

(b) The same superadditive pattern.

(c) Left: From Liu et al. (2019). Right: The same positive correlation.

Figure 2: The top row shows (a) the modeled judgment patterns, matching empirical judgment patterns, from Pearl & Sprouse (2013), and (b) the judgment patterns (log probabilities) generated by the Fragment Grammar (FG) identified by the modeled learner, given realistic samples of child-directed speech. The bottom row shows (c) left: the empirical judgment patterns from Liu et al. (2019), and right: the judgment patterns generated by the FG.

Hastings algorithm to iteratively resample a potential FG representation for each item in the input and then accepts or rejects the representation to increase the probability of the input data.

To approximate the *wh*-dependency input that children learn from, we collected 12,704 *wh*-dependencies from the CHILDES Treebank (Pearl and Sprouse, 2013) and extracted the dependency path from each.[1] We then estimated the counts of the dependencies that children would encounter by four years old, when some syntactic island knowledge seems to be present (De Villiers et al., 2008).[2] From this input, the modeled learner infers the best fragments for the *wh*-dependencies in its input, which may or may not include lexically-subcategorized phrasal structures for any given fragment. This model allows us to explore all the possibilities of lexically subcategorizing different phrasal categories as opposed to implementing a particular hypothesis (i.e. main verbs are always lexicalized or CPs are always lexicalized).

We find that the learned FG dependency representation can be used to correctly generate all previously-attested acceptability judgment patterns

[1] See Supplemental Section A.1 for more details.

[2] We drew on estimations by Bates and Pearl (2021) that consider waking hours, utterances per hour, and *wh*-dependency frequency in children's input between 20 months (when *wh*-dependencies are reliably processed) and 4 years.

(Figure 2b and d).[3] Notably, the FG representation's fragments lexically subcategorize phrasal structures only for some more-frequent items (e.g., VP_{think}, CP_{that}, VP_{say}-CP). This means the modeled learner automatically determined the best frequency threshold for lexically subcategorizing each individual phrase structure type, due to the goal of efficient representation.

5 Comparison representations

We compared the FG representation's performance against several trigram-based baseline representations (2), all of which used the same input as the FG model.[4] We chose trigram-based representations, as n-grams are common representations in language modeling (see Manning and Schutze 1999 for a review), and trigram-based representations have been used in prior successful models that predict adult judgement patterns of *wh*-dependencies (Pearl and Sprouse, 2013). A trigram-based representation also can (i) be paired with a straightforward learning algorithm (e.g., tracking frequencies of the trigrams in the input), and (ii) can transparently reflect different proposals for lexical subcategorization, as in (2).

(2) Baseline trigam representations

a. no-lexicalization: phrase labels only, e.g., "IP-VP-CP"

b. fully-lexicalized: subcategorized phrase labels, e.g., "IP_{past}-VP_{claim}- CP_{that}"

c. CP-lexicalized (from Pearl and Sprouse 2013): only CP is subcategorized, e.g., "IP-VP-CP_{that}"

d. main-V-lexicalized (in line with Liu et al. 2019): only main V is subcategorized, e.g., "IP-VP_{claim}-CP"

We selected the no-lexicalization and the fully-lexicalized representations as the two extremes of our hypothesis space; we can include no lexical information or all the lexical information for phrase structure nodes in a trigram-based dependency representation. The remaining two representations each implement a hypothesis about what lexical information should be included in the phrasal structure, inspired by previous work: the CP-lexicalized representation from Pearl and Sprouse (2013), and the main-V-lexicalized representation from Liu et al. (2019).

Most baselines failed to capture the full range of acceptability judgment patterns: the no-lexicalized failed to capture Adjunct and Whether islands, as well as the verb frequency effect; the fully-lexicalized failed to capture Adjunct islands; and the CP-lexicalized failed to capture the verb frequency effect. However, the main-V-lexicalized did capture all the acceptability patterns. We note that the FG representation also lexicalized main verbs (though only those that were more frequent), and so has this in common with the main-V-lexicalized baseline (which lexicalized all main verbs, irrespective of frequency). We note that one advantage of the inferred FG representation over the main-V-lexicalized representation is that the FG representation was automatically learned – including which parts are lexicalized and how large the pieces are that comprise a dependency path – rather than needing to be specified beforehand, as the trigram-based main-V-lexicalized baseline was.

6 Conclusion

Here we have explored how children could learn constraints on English *wh*-dependencies by focusing their learning efforts on how to efficiently represent *wh*-dependencies, rather than trying to explicitly learn the constraints. The specific approach we explored involved a modeled learner attempting to identify the best Fragment Grammar (FG) for efficiently representing the *wh*-dependencies encountered in English child-directed speech. The FG representation allowed the modeled learner to generate all the acceptability judgement patterns previously attested to reflect knowledge of different constraints on *wh*-dependencies, known as syntactic islands. Because the modeled learner learned from input that four-year-olds would encounter, one testable prediction that future behavioral work can investigate is that four-year-olds should in fact have acquired all the syntactic knowledge assessed via the acceptability judgment patterns used here if four-year-olds are in fact using an FG representation. Additionally, future work can investigate predictions for other *wh*-dependency constraints known to be acquired by children around age four (De Villiers et al., 2008), comparing the FG representation against other representational possibilities, such as the main-V-lexicalized baseline.

[3] See Supplemental Section A.4 for details.

[4] These baselines additionally had a "Start" and "End" symbol in their dependency paths to ensure each dependency created at least one trigram. For instance, a main clause subject dependency like "What happened?" would be represented with the trigram "Start-IP-End".

References

Alandi Bates and Lisa Pearl. 2021. When do input differences matter? using developmental computational modeling to assess input quality for syntactic islands across socio-economic status.

Noam Chomsky. 1973. Conditions on transformations. In S. Anderson and P. Kiparsky, editors, *A Festschrift for Morris Halle*, pages 237–286. Holt, Rinehart, and Winston, New York.

Jill De Villiers, Thomas Roeper, Linda Bland-Stewart, and Barbara Pearson. 2008. Answering hard questions: Wh-movement across dialects and disorder. *Applied Psycholinguistics*, 29(1):67–103.

Yingtong Liu, Rachel Ryskin, Richard Futrell, and Edward Gibson. 2019. Verb frequency explains the unacceptability of factive and manner-of-speaking islands in english. In *CogSci*, pages 685–691.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Timothy O'Donnell, Jesse Snedeker, Joshua Tenenbaum, and Noah Goodman. 2011. Productivity and reuse in language. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33.

Timothy J O'Donnell. 2015. *Productivity and reuse in language: A theory of linguistic computation and storage*. MIT Press.

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*, 20:19–64.

A Supplemental Material

A.1 Preprocessing from CHILDES

Using NLTK and Python, we extracted the *wh*-dependency trees from the CHILDES Treebank. Using the trace annotations in the corpora, we extracted the path from the gap position to the *wh*-word, including the phrase label (e.g., VP) and its lexical child (e.g., *think*) in the resulting sequences. When a VP followed the IP in a dependency path, tense was added as the lexical child of IP nodes (e.g., *thought* would yield IP$_{past}$-VP$_{think}$).

A.2 Preprocessing

Preprocessing for the FG grammar input: We created the tree structures of the dependency paths from these sequences in a form that the FG learner can process (e.g. the *wh*-dependency "What are you eating?" would be encoded as "((IP (LEX present) (VP (LEX eat))))"). We note also that IP-only dependencies like "*What happened?*" did not have the IP lexicalized with tense (e.g., the FG input representation would be (IP null)).

Preprocessing for the trigram baseline models: The baseline trigram models took the final dependency path, extracted from CHILDES and preprocessed to include tense (e.g., IP$_{past}$-VP$_{think}$), and extracted appropriate trigrams, depending on the baseline. We included a "Start" and "End" symbol in our dependency paths for the baselines in order for all paths to be one trigram at a minimum. This allowed IP-only dependencies to be handled by trigram-based models (i.e., the trigram would be Start-IP-End).

A.3 Inference of the best FG grammar

The inference algorithm used to identify the best FG was implemented using code provided by Tim O'Donnell. We used the default parameter values: pitman-Yor (PY) *a* set to 0 and PY *b* set to 1; sticky concentration parameter set to 1 and sticky distribution parameter set to 0.5; the Dirichlet-multinomial pseudo-counts (pi parameter) were set to 1; the model performed 1000 sweeps.

A.4 Generating predictions of acceptability using the FG representation

When generating predictions for *wh*-dependencies, based on the FG representation, we extracted the same form of the *wh*-dependency path from the Pearl and Sprouse (2013) and Liu et al. (2019) stimuli. Due to the design of the code, structures that required rules the FG did not hypothesize would yield no output (i.e., cause a code crash). To circumvent this and be able to generate predictions for structures like those that cross syntactic islands, we needed to add all possible phrase rules to the FG representation. So, we added all possible rules (in the form "Label1 – Label2 Label3", where a Label was a phrase structure node like IP or PP) that the FG representation did not create through inference. We then gave these rules "counts" of 0.5 (as opposed to any seen structure having a count of at least 1) and re-normalized the log probabilities of all rules.

Masked language models directly encode linguistic uncertainty

Cassandra L. Jacobs
Department of Linguistics
University at Buffalo
cxjacobs@buffalo.edu

Ryan J. Hubbard & Kara D. Federmeier
Beckman Institute and Department of Psychology
University of Illinois at Urbana-Champaign
{rjhubba2;kfederme}@illinois.edu

1 Introduction

Recent advances in human language processing research have suggested that the predictive power of large language models (LLMs) can serve as cognitive models of human language processing. Evidence for this comes from LLMs' close fit to human psychophysical data, such as reaction times or brain responses in language comprehension experiments. Those adopting LLM architectures as models of human language processing frame the problem of language comprehension as prediction of the next linguistic event (Goodkind and Bicknell, 2018; Eisape et al., 2020), in particular focusing on lexical or syntactic surprisal. However, this approach fails to consider that comprehenders make predictions using some representation of the *content* of an utterance. That is, in contrast to surprisal, readers make use of a mental model that creates an evolving understanding of who is doing what to whom and how. In contrast to comprehenders, surprisal measures do not make predictions about the content, as surprisal simply measures the conditional probability of some linguistic event given the surrounding context.

Many convergent cues in the upstream context, such as the frequencies of words in a sentence so far, will affect hidden state representations of models, which may then influence the predictability of upcoming words. The present work deviates from the surprisal paradigm by assessing how much the hidden state representations of LLMs, which are the source of the predictive power that LLMs have over symbolic representations, encode human language processing-relevant uncertainty. We specifically assess this possibility using the stimulus set from Federmeier et al. (2007), which contains sentences that manipulated the predictability of a final word by designing the sentences to be either strongly or weakly constraining. We therefore sought to test whether it is possible to predict constraint from the sentence embeddings directly to better understand whether and how linguistic uncertainty is encoded in hidden states.

2 Cloze completion dataset

We constructed a cloze completion dataset (Taylor, 1953) to compare one LLM (RoBERTa; (Liu et al., 2019)) to human predictions of final words in Federmeier et al. (2007). This stimulus set contains 282 sentence stimuli that differ in the **constraint** of the sentence, or the degree to which the preamble leads the reader in a specific direction. Broadly, *strongly* constraining sentences consistently lead readers in one semantic direction and the final word (**critical**) is highly predictable; *weakly* constraining sentences are less specific, which we summarize below.

- *Strong Constraint*: Sharon dried the bowls with a **towel**.
- *Weak Constraint*: He always seemed to be interested in looking at the **sky**.

One property of these stimuli is that constraint is partly defined by the predictability of the final word. Cloze probability is defined as the percentage of completions produced by participants that end in a particular final word. For example, if *towel* is guessed by 30% of participants, then its cloze probability is .3. Effects of constraint have sometimes been assessed by categorizing sentences using the cloze probability of the most expected completion, as in Federmeier et al. (2007) [strong constraint: cloze $>$ 67%; weak constraint: cloze $<$ 42%]. Constraint is therefore both a product of the vagueness or specificity of the preamble, and the predictability of the (**bolded**) critical word given the preamble.

The present cloze dataset includes 109,225 word-by-word predictions for all non-initial words from 158 participants recruited from the Prolific platform. All participants self-reported as having acquired American English before age 5 and received

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 225-228.
Held on-line February 7-9, 2022

$8 for 30 minutes of their time. Here we are interested in whether the preamble encodes the predictability (constraint) of the final word.

In general, the cloze probability when participants produced the intended final word from Federmeier et al. (2007) was higher when the sentence was strongly constraining (SC; $\hat{\mu}_{SC} = 0.71$) than when it was weakly constraining (WC; $\hat{\mu}_{WC} = 0.21$), $t(266) = 25.1$, $p < .001$. We therefore largely replicated the original divisions of Federmeier et al. (2007). However, the current cloze dataset differs from the original stimulus set in that we are able to leverage the probabilities of all cloze completions to assess uncertainty across these categories. Participants provided more varied responses as evidenced by higher entropy in WC sentences ($\hat{\mu}_{WC} = 0.87$) than in SC ones ($\hat{\mu}_{SC} = 0.36$), $t(200) = 21.8$, $p < .001$, a result we discuss further in Section 3.2. In the next section, we describe our masking procedure for assessing the degree to which cloze probabilities and response entropies correlate with embedding representation-derived measures.

3 Probing the predictability of final words

Given the clear difference in cloze probabilities of critical words in the strongly and weakly constraining stimuli in Section 2, we reasoned that strongly and weakly constraining sentences are relatively easy for participants to distinguish. In this section, however, we sought to test whether the unpredictability of a word as defined by the original cloze labels in Federmeier et al. (2007) is recoverable directly from sentence embeddings, as outlined in Section 2. While this may seem trivial, it is not obvious exactly what factors influence the predictability of a final word – individually or jointly. For example, it is possible that comprehenders rely predominantly on immediately preceding information when completing cloze tasks, but they may also incorporate linguistic properties of words or combinations of words earlier in the sentence (MacDonald and Seidenberg, 2006).

To test whether constraint is recoverable from sentence embeddings, we leveraged the masked language model RoBERTa (Liu et al., 2019), which enabled us to hide the critical words from the model's representation of the sentence and obtain sentence embeddings for a downstream probing model. RoBERTa deviates from human language processing in that it processes the entire sentence simultaneously, rather than incrementally as in recurrent neural networks (Elman, 1990). However, we can present sentences except the final word to RoBERTa, which can mimic any forward predictions and higher-order integration that readers will have done up until that point. Importantly, a masked language model like RoBERTa allows us to mask the final word, and obtain a representation of only the upstream (preamble) part of the sentence.

We then transformed the sentence into a single vector for our classification procedure, taking the original sentence from the Federmeier et al. (2007) stimuli, except we replaced the critical final word with a `<mask>` token. Embedding the sentence using RoBERTa produces a fixed-length vector for each token (roughly, word), from which we computed a sentence embedding vector by averaging all token vectors within each layer, excluding the `<mask>` token. This embedding process produced a $282 \times 13 \times 768$-dimensional matrix. From these embeddings, we then constructed 282 leave-one-out regularized logistic regression probing classifiers (one for each critical sentence) trained on 281 of the sentence embeddings to predict strong (SC) or weak constraint (WC) from the original Federmeier et al. (2007) labels. We then treat the remaining sentence as a test item and obtain a predicted probability of the sentence being strongly constraining.

3.1 Cloze surprisal

In contrast to using raw percentages of completions of the Federmeier et al. (2007) cloze stimuli, we can alternately quantify constraint using either the surprisal of a particular completion (Eq. 1) or estimate entropy (H; Eq. 2) over all K cloze completions:

$$\text{surprisal} = -\log(p(x)) \tag{1}$$

$$H = \sum^{K} p(x) \cdot \log_K(p(x)) \tag{2}$$

If constraint is encoded in both the final resulting sentence and the context, then we expect to see a positive relationship between the model's belief that the sentence is strongly constraining and participants' ability to guess the target word. However, constraint may also be measured using cloze probabilities, or the conditional probability of participants producing a word given a context. In the Federmeier et al. (2007) work, strongly and weakly constraining sentences were designed to have high and low cloze probability completions, respectively. Therefore, we tested for a correlation between linguistic uncertainty as estimated by the cloze probability of the critical word and the predicted probabilities obtained from the classifiers.

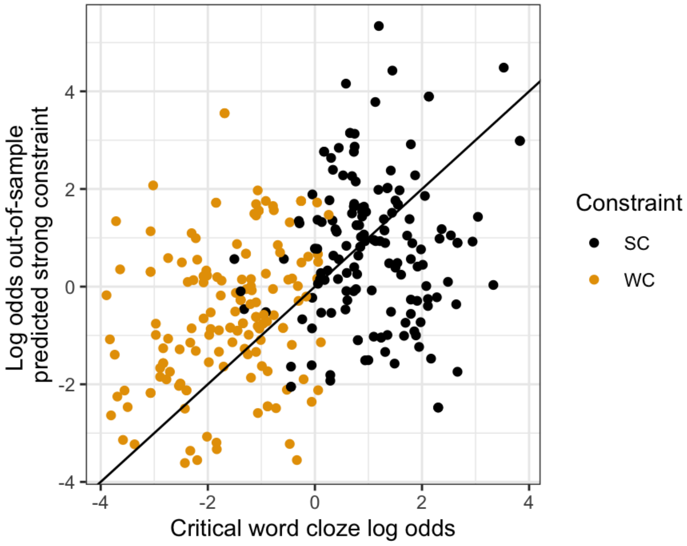


Figure 1: Plotted relationship between critical final word cloze and classifier probability of constraint ($\hat{\rho} = 0.43$, $p < .001$). Line represents perfect correlation.

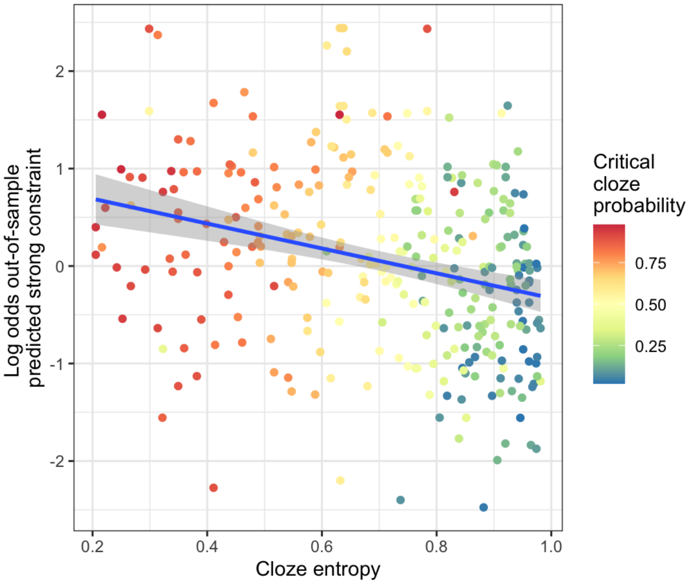


Figure 2: Plotted relationship between final word entropy and classifier probability of constraint. Line represents fitted slope ($\hat{\rho} = -.32$, $p < .001$).

With these predicted probabilities, we then tested for a relationship between the log odds of a predicted SC label as a function of the cloze probability of the final completion and found a strong correlation between the two ($\hat{\rho} = 0.43$, $p < .001$). We plot this relationship in Figure 1.

3.2 Cloze entropy

Like cloze probability, we can also compute the uncertainty of participants' final responses by computing the entropy of the outcomes. This uncertainty captures the intuition that if participants vary in what they expect, then their guesses will be relatively uniformly distributed across many outcomes. Indeed, the weakly constraining sentences in Federmeier et al. (2007) may have been designed to be vague, and thus intended to be completed by many possible valid words. We conducted the same analysis as in the previous section, and found that with greater uncertainty (higher entropy), the model's belief that a sentence was strongly constraining decreased ($\hat{\rho} = -.32$, $p < .001$). We plot this relationship in Figure 2. This strongly suggests that RoBERTa encodes final word uncertainty in a similar way to how it encodes constraint.

4 Discussion

In two experiments we have tested how much context on its own – without knowledge of the final word – can directly encode the predictability of upcoming linguistic information. In contrast to prior work focusing on surprisal, this work leverages experimenter-defined labels (sentential constraint categories) and sentence embeddings derived from the LLM RoBERTa and shows that the model's hidden states directly encode uncertainty about upcoming information. We demonstrated that we are able to train classifiers that can predict the categorical constraint of a sentence and that the model's certainty about the constraint category correlates with the cloze probability of the target word and relatedly the entropy of participants' responses.

These results present an interesting puzzle about how lexical predictability unfolds in human language comprehension. For example, readers build up representations of sentences incrementally as they read through a sentence, though they may read back in a passage or reread some sections of text. In turn, this higher-order representation of the language guides their expectations about upcoming words (Lowder et al., 2018), one aspect of which may be uncertainty or the semantic specificity of predictions that can be made.

In sum, we have presented one of the first attempts at using embeddings instead of computing surprisal values to account for the lexical predictability of words in sentences. We believe that the method outlined here raises several questions about how predictions are launched and how uniformly throughout utterances vagueness or uncertainty is encoded. These questions include topics that are critical from a multiple constraint satisfaction approach (MacDonald and Seidenberg, 2006), such as which words contribute the most toward the predictions of the final words. In future work, we hope to also analyze non-final word uncertainty using similar methods to better understand how cloze probabilities relate to sentence representations as the sentence unfolds. Analyses of attention patterns in LLMs (e.g., Vig and Belinkov, 2019) and masking of specific words may provide some clues to the sources of predictions.

References

Tiwalayo Eisape, Noga Zaslavsky, and Roger Levy. 2020. Cloze distillation: Improving neural language models with human next-word prediction. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 609–619, Online. Association for Computational Linguistics.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Kara D Federmeier, Edward W Wlotko, Esmeralda De Ochoa-Dewald, and Marta Kutas. 2007. Multiple effects of sentential constraint on word processing. *Brain Research*, 1146:75–84.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Matthew W Lowder, Wonil Choi, Fernanda Ferreira, and John M Henderson. 2018. Lexical predictability during natural reading: Effects of surprisal and entropy reduction. *Cognitive science*, 42:1166–1183.

Maryellen C MacDonald and Mark S Seidenberg. 2006. Constraint satisfaction accounts of lexical and sentence comprehension. In *Handbook of psycholinguistics*, pages 581–611. Elsevier.

Wilson L Taylor. 1953. "Cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

MaxEnt Learners are Biased Against Giving Probability to Harmonically Bounded Candidates

Charlie O'Hara
Department of Linguistics
Univeristy of Michigan
cohara@umich.edu

1 Overview

One of the major differences between MaxEnt Harmonic Grammar (Goldwater and Johnson, 2003) and Noisy Harmonic Grammar (Boersma and Pater, 2016) is that in MaxEnt harmonically bounded candidates are able to get some probability, whereas in most other constraint-based grammars they can never be output (Jesney, 2007). The probability given to harmonically bounded candidates is taken from other candidates, in some cases allowing MaxEnt to model grammars that subvert some of the universal implications that are true in Noisy HG and categorical forms of HG (Anttila and Magri, 2018). Magri (2018) argues that the types of implicational universals that remain valid in MaxEnt are phonologically implausible, suggesting that MaxEnt overgenerates Noisy HG in a problematic way.

However, a variety of recent work has shown that some of the possible grammars in a constraint based grammar may be unlikely to be observed because they are difficult to learn (Staubs, 2014; Stanton, 2016; Pater and Moreton, 2012; Hughto, 2019; O'Hara, 2021). Here, I show that grammars that give too much weight to harmonically bounded candidates, and violate the implicational universals that hold in Noisy HG are significantly harder to learn than those grammars that are also possible in Noisy HG. With learnability applied, I claim that the typological predictions of MaxEnt and Noisy HG are in fact much more similar than they would seem based on the grammars alone. This paper focuses on the classically harmonically bounded candidates, because collectively bounded candidates reflect a different type of constraint weighting, and are more often observed typologically (see local optionality Riggle and Wilson (2005); Hayes (2017)).

2 The Problem

Anttila and Magri (2018) show that MaxEnt overpredicts Noisy HG. Specifically, given a specific set of constraints, there are probabilistic universals in Noisy HG that are not maintained in MaxEnt; in other words for all Noisy HG grammars the probability of one mapping (/x/→[y]) is always less than or equal to the probability of some other mapping (/$\hat{x}$/→[$\hat{y}$]), but in MaxEnt the former mapping can be more probable. They characterize the difference between MaxEnt and Noisy HG geometrically, showing that the probabilistic universals generated by Noisy HG are a superset of those generated by MaxEnt for any particular set of tableaux.

Figure 1: In order for a particular mapping /x/→[y] to be always assigned a lower or equal probability than the mapping /$\hat{x}$/→[$\hat{y}$]: in Noisy HG all difference vectors between /$\hat{x}$/→[$\hat{y}$] and its competitors must fall in the dashed region, whereas in MaxEnt, they must fall in the crosshatched region. The dots represent the difference vectors of /x/→[y] compared to its competitors. Adapted from Anttila and Magri (2018).

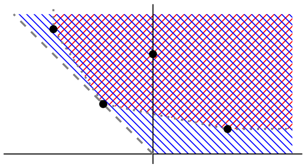


Figure 1 shows an example of this difference in a system with two constraints. Each node represents a difference vector between the antecedent mapping /x/→[y] and one of its competitors /x/-[z], calculated by subtracting the violations of /x/→[z] from /x/→[y] (assuming violations are counted negatively). Anttila and Magri (2018) show that in order for some consequent mapping /$\hat{x}$/→[$\hat{y}$] to never receive a lower probability than the antecedent /x/→[y] mapping under all weightings of constraints, all difference vectors between /$\hat{x}$/→[$\hat{y}$] and its competitors must have fall in the region greater than the convex hull generated by the antecedent difference vectors in MaxEnt (correspond-

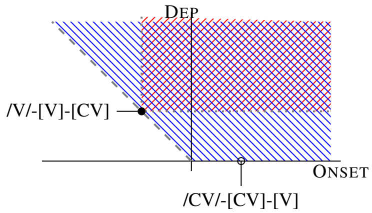
Table 1: Universal-Subverting Pattern in MaxEnt

/CV/	ONSET	DEP		
Weights	$w = 2$	$w = 5$	HARM	PROB
a. CV			0	0.88
b. V	-1		-2	0.12
/V/	ONSET	DEP		
Weights	$w = 2$	$w = 5$	HARM	PROB
a. CV		-1	-5	0.05
b. V	-1		-2	0.95

ing to the crosshatched region).[1] In Noisy HG, the consequent's difference vectors can fall anywhere in the region greater than the convex *cone* generated by the antecedent difference vectors (also including the dashed regions). Here, I will argue that many of these cases are caused by the fact that MaxEnt assigns probability to harmonically bounded candidates but Noisy HG does not.

A simple concrete example emerges in syllable structure—using the constraints and candidates in Table 1, it is quite obvious in noisy HG that onsetful syllables map faithfully (/CV/-[CV]) at least as often as onsetless syllables do (/V/-[V]), since /CV/-[CV] harmonically bounds its competitor. However, in MaxEnt it is possible for the onsetless faithful mapping to receive more probability than the onsetful mapping, see Table 1.[2] This difference between MaxEnt and Noisy HG is directly caused by the harmonically bounded candidate /CV/-[V] being able to take probability from the /CV/-[CV] mapping only in MaxEnt. This type of *classically harmonically bounded* candidate can only receive any probability when the bounding constraints (here MAX and ONSET) are sufficiently low-weighted. This difference is geometrically represented in Figure 2. The filled dot represents the difference vector between /V/→[V] and /V/→[CV], whereas the unfilled dot represents the difference vector between /CV/→[CV] and /CV/→[V]. Crucially, the unfilled dot falls only in the dashed region, but not the crosshatched region.

Figure 2: Geometric representation of the onset typology with DEP and ONSET.



Harmonically bounded candidates show particular geometric properties. A harmonically bounded candidate violates a superset of the violations of some other candidate. If the candidate is bounded by the target mapping, the difference vector between the target vector and the candidate will be non-negative for all constraints, placing it in the first quadrant (top right) of the graph. If a candidate is harmonically bounded by some other candidate, it will be at least as large (component by component) than the candidate that harmonically bounds it. The second case is less problematic in MaxEnt because if /x/-[y]-[z] harmonically bounds /x/-[y]-[$\hat{z}$], and the difference vector for [$\hat{z}$] falls outside of the cross hatched region for some antecedent vector, so must the difference vector for [z]. On the other hand, as seen in the example above, when the target mapping harmonically bounds a candidate, that candidate can fall in the first quadrant, but below the convex hull generated by the set of antecedent difference vectors. We can see that a large portion of the difference vectors that behave differently in MaxEnt and Noisy HG are of this subtype—they fall in the region in the first quadrant under the convex hull.[3] Harmonically bounded candidates only receive probability under certain restricted weighting conditions—as the weight of the harmonically bounded constraints increases, the probability assigned to candidates bounded by those constraints becomes vanishingly small. If not all weighting conditions are equally easy to learn, is it possible that it is particularly hard to learn constraint weightings that would assign a significant probability to harmonically bounded candidates?

[1] As long as the number of competitors for the antecedent and consequent are the same.

[2] So that this system can be represented two-dimensionally, here I am excluding MAX, as well any constraints or candidates with codas. These will be introduced later in the paper for the simulations. This situation is the same as if MAX was weighted zero, and NOCODA was weighted very high.

[3] There are two other regions that differentiated MaxEnt and Noisy HG—in this two-dimensional representation, the triangle generated by the origin, the y-axis and the left edge of the convex cone, and the triangle generated by the leftmost difference vector, the left edge of the cone, and the left edge of the region larger than the convex hull. I save characterization of these regions for future work.

2a. Categorical Pattern				
	Output			
Input	[CV]	[V]	[CVC]	[VC]
/CV/	1	0	0	0
/V/	0	1	0	0
/CVC/	0	0	1	0
/VC/	0	0	0	1

2b. Universal Respecting Pattern				
	Output			
Input	[CV]	[V]	[CVC]	[VC]
/CV/	1	0	0	0
/V/	.5	.5	0	0
/CVC/	.5	0	.5	0
/VC/	.25	.25	.25	.25

2c. Universal Subverting Pattern				
	Output			
Input	[CV]	[V]	[CVC]	[VC]
/CV/	.5	.5	0	0
/V/	0	1	0	0
/CVC/	.25	.25	.25	.25
/VC/	0	.5	0	.5

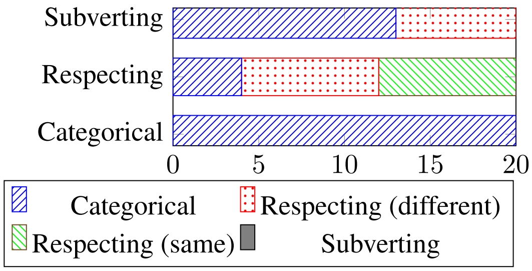
Table 2: Patterns under consideration

3 Learnability

To evaluate the learnability of different classes of grammars, I make us of agent-based generational learning simulations (Kirby and Huford, 2002; Kirby, 2017). These simulations make use of a series of learning agents using the Perceptron learning algorithm (Rosenblatt, 1958; Jäger, 2003; Boersma and Pater, 2016); each initialized following conventional assumptions in the phonological learning literature (i.e. markedness constraints weighted high faithfulness low (Gnanadesikan, 2004; Tesar and Smolensky, 2000; Jesney and Tessier, 2011)). Learners are exposed to a limited number of input-output mappings randomly chosen from their target grammar (each underlying syllable type is sampled equally frequently, surface forms sampled according to the target grammar). After the learner is exposed to the number of forms (here 7000 forms per generation), the learner *matures* and whatever grammar it learned is used as the target grammar for the next learner. Each run of the simulation consists of 15 generations, with the first generation exposed to whatever grammar is being tested.

Three types of patterns were tested: one fully categorical pattern available in MaxEnt and Noisy HG (2a), one variable grammar that is consistent with the implicational universals (2b), and one variable grammar that subverts the implicational universals (2c). Notably, only the last pattern gives any probability to harmonically bounded candidates.

Figure 3: Resulting patterns after 15 generations.



4 Simulation Results

The simulations show that the categorical patterns are learned most consistently, followed by the universal-respecting variation patterns. The universal-subverting patterns available only in MaxEnt are learned consistently worse than the other types of patterns on multiple metrics. First, the universal subverting patterns require much more data to be learned accurately, as shown by the number of iterations it took to learn the pattern on average in the first generation (Table 3). Further we can look the end result of the 20 runs performed for each simulation to see how stably the pattern is learned across generations, which allows us to see how likely a pattern is to change, and how likely a pattern is to be innovated. Figure 3 presents the results after fifteen generations, classified according to what the initial target pattern was, and what the pattern the final generation learned would be classified as. It can be seen that the categorical pattern is learned fully stably under these parameters; whereas the universal respecting variation changes in 12 of the 20 runs, often reducing the variability of the pattern. Finally, the universal subverting patterns are learned very unstably, changing into a type of pattern that can be modeled in Noisy Harmonic Grammar in all 20 runs.

Table 3: # of iterations needed to learn each pattern.

Grammar Type	Iterations Needed
Categorical	2000
Respecting	2200
Subverting	5000

Figure 4: 100 learners trained on normal variation with 60% coda deletion.

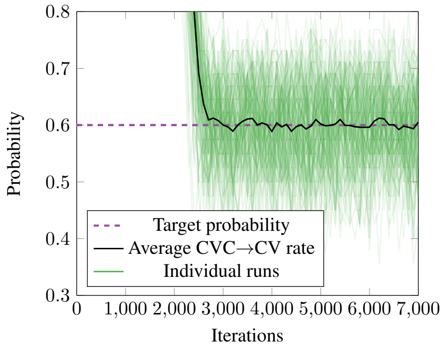
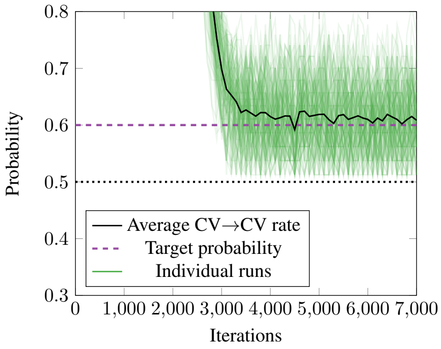


Figure 5: 100 learners trained on harmonically bounded variation with 40% onset deletion



5 Discussion

The universal-subverting patterns are harder to learn because it is necessary for the weights of some constraints to approach zero, rather than simply becoming lower or higher than some conflicting constraint. In this case, the only evidence that would force a constraint close to zero is from observing harmonically bounded candidates in the target grammar. The difficulty learners have learning typology subverting patterns is due to the convergence properties of online MaxEnt learner that restrict constraint weights to non-negative numbers. While this learning algorithm is weakly convergent (Fischer, 2005), I show that the expected weighting of a learner upon convergence differs from the target weighting substantially more when that target weighting has constraint weights close to zero—a necessary property of typology subverting variation patterns, but not typology respecting variation.

When learning a variable pattern, individual learners do not ever stop updating, because even if the learner and teacher have the same grammar, errors still occur. Each individual learner ends up oscillating around the target pattern. When this variation is symmetrical, the average across many learners converges to the target pattern. However, when the target pattern requires a constraint being weighted particularly close to zero, learners oscillate asymmetrically—some learners

This learning bias is of a stronger sort than many considered in the learning literature, rather than simply requiring more time to converge, learners trained on typology subverting patterns converge on a grammar different from the target grammar. To demonstrate a basic example of how the learning algorithm converges more accurately to normal variation than harmonically bounded variation, I ran 100 learners on two variable patterns. In the normal variation pattern, onset consonants neither epenthesized or deleted (100% faithful) and coda consonants deleted 60% of the time. In the harmonically bounded variation pattern, coda consonants deleted categorically, but onsets deleted 40% of the time. Each simulation ran according to the parameters of the above simulations. Figures 4 and 5 show the results of these simulations. The dark black line represents the average probability of the target variable mapping across all 100 learners, whereas the lighter green lines represent each individual run. The dashed gray line shows the target probability of the mapping. In normal variation (Figure 5), the learners oscillate symmetrically around the target pattern, with the average staying very close to the target probability. In harmonically bounded variation (Figure 6), the average remains notably above the target probability. Harmonically bounded variation acts differently because learners cannot oscillate symmetrically around the target pattern—learners assigning less probability to the target mapping end up "bouncing" off of a wall, because the harmonically bounded CV→V mapping can never receive more than 50% because constraint weights must remain nonnegative.

If phonological learners are biased against assigning probability to harmonically bounded candidates even when weightings exist in MaxEnt that assign probability to them, a major source of typological difference between MaxEnt and Noisy HG appears to be less significant. Future work will investigate the other geometrical regions of difference between MaxEnt and Noisy HG, and explore whether they also require very low constraint weights that are difficult to learn.

Acknowledgments

Thanks to Karen Jesney, Caitlin Smith, Rachel Walker and audiences at USC's PhonLunch, AMP 2019, and University of Michigan's Phondi meetings for comments on this project.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Arto Anttila and Giorgio Magri. 2018. Does maxent overgenerate? implicational universals in maximum entropy grammar. In *Proceedings of the 2017 Annual Meeting on Phonology*. Linguistics Society of America.

Paul Boersma and Joe Pater. 2016. Convergence properties of a gradual learning algorithm for Harmonic Grammar. In John J. McCarthy and Joe Pater, editors, *Harmonic Grammar and Harmonic Serialism*. Equinox.

Benjamin Börschinger and Mark Johnson. 2011. A particle filter algorithm for Bayesian wordsegmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Markus Fischer. 2005. A Robbins-Monro type learning algorithm for an entropy maximizing version of stochastic optimality theory. Master's thesis, Humboldt University, Berlin.

Amalia Gnanadesikan. 2004. Markedness and faithfulness in child phonology [ROA-67]. In René Kager, Joe Pater, and Wim Zonneveld, editors, *Fixing Priorities: Constraints in Phonological Acquisition*, pages 73–108. Cambridge University Press, Cambridge.

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a Maximum Entropy model. In *Proceedings of the Workshop on Variation within Optimality Theory*. Stockholm University.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Bruce Hayes. 2017. Varieties of noisy harmonic grammar. In *Proceedings of the 2016 Annual Meeting on Phonology*, Washington, DC. Linguistics Society of America.

Coral Hughto. 2019. *Emergent Typological Effects of Agent-based learning models in Maximum Entropy Grammar*. Ph.D. thesis, University of Massachusetts Amherst.

Gerhard Jäger. 2003. Learning constraint sub-hierarchies: the bidirectional gradual learning algorithm. In Henk Zeevat and Reinhard Blutner, editors, *Optimality Theory and pragmatics*, pages 251–287. Palgrave Macmillan, Basingstoke.

Karen Jesney. 2007. The locus of variation in weighted constraint grammars. Handout for poster presented at the Workshop on Variation, Gradience and Frequency in Phonology, Stanford University.

Karen Jesney and Anne-Michelle Tessier. 2011. Biases in harmonic grammar: The road to restrictive learning. *Natural Language & Linguistic Theory*, 29.

Simon Kirby. 2017. Culture and biology in the origins of linguistic structure. *Psychonomic Bulletin and Review*, 24:118–137.

Simon Kirby and James Huford. 2002. The emergence of linguistic structure: An overview of the iterated learning model. In A Cangelosi and D. Parisi, editors, *Simulating the Evolution of Language*, chapter 6, pages 121–148. Springer Verlag, London.

Giorgio Magri. 2018. Implicational universals in stochastic constraint-based phonology implicational universals in stochastic constraint-based phonology implicational universals in stochastic constraint-based phonology. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Charlie O'Hara. 2021. *Soft Biases in Phonology: Learnability meets grammar*. Ph.D. thesis, University of Southern California.

Joe Pater and Elliott Moreton. 2012. Structurally biased phonology: complexity in language learning and typology. *The EFL Journal*, 3(2):1–44.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Jason Riggle and Colin Wilson. 2005. Local optionality. In *Proceeding of the North Eastern Linguistics Society*, volume 35.

F Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

Juliet Stanton. 2016. Learnability shapes typology: the case of the midpoint pathology. *Language*, 92(4):753–791.

Robert Staubs. 2014. *Computational modeling of learning biases in stress typology*. Ph.D. thesis, University of Massachusetts Amherst, Amherst.

Bruce Tesar and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.

Universal Dependencies and Semantics for English and Hebrew Child-directed Speech

Ida Szubert
University of Edinburgh
k.i.szubert@sms.ed.ac.uk

Omri Abend
Hebrew University of Jerusalem
omri.abend@mail.huji.ac.il

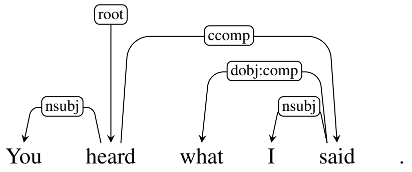
Nathan Schneider
Georgetown University
nathan.schneider@georgetown.edu

Samuel Gibbon **Sharon Goldwater** **Mark Steedman**
University of Edinburgh
{samuel.gibbon@, sgwater@inf., steedman@inf.}ed.ac.uk

1 Introduction

While corpora of child speech and child-directed speech (CDS) have enabled major contributions to the study of child language acquisition, semantic annotation for such corpora is still scarce and lacks a uniform standard. We compile two CDS corpora—in English and Hebrew—with syntactic and semantic annotations. We employ a methodology that enforces a cross-linguistically consistent representation, building on recent advances in dependency representation and semantic parsing. Our semi-automatic syntactic annotation follows the Universal Dependencies standard (UD; de Marneffe et al., 2021), adapted to suit the CDS genre. To induce semantic forms, we develop an automatic method for transducing UD structures into sentential logical forms (LFs), e.g. figure 1. The two representations have complementary strengths: UD structures are language-neutral and support direct annotation, whereas LFs are neutral as to the syntax-semantics interface, and transparently encode semantic distinctions.

What follows is a brief synopsis of the work, which is described in full in (Szubert et al., 2021).



LF: $\lambda e_1.\ heard_{e_1}(you,\ \text{WHAT}\ x\big[\lambda e_2.\ said_{e_2}(I, x)\big])$

Figure 1: Example syntactic and semantic annotation.

2 Related Work

The CHILDES project (MacWhinney, 2000) has been pivotal in efforts to streamline linguistic data collection of child–caregiver interactions and to standardize linguistic annotation in this domain. CDS resources annotated with *semantic* annotation, however, are scarce, and lack a uniform standard. Indeed, even *syntactic* annotation is only available in CHILDES for a handful of languages, and these are not all annotated according to the same scheme.

Syntax. To the extent that CHILDES data has been syntactically annotated, various syntactic representations have been adopted (Sagae et al., 2010; Pearl and Sprouse, 2013; Odijk et al., 2018). Given the state of the art in multilingual parsing, we argue that UD is the best choice for cross-linguistically comparable syntactic annotation.[1]

Semantics. Sentential logical forms (henceforth, LFs) are an essential building block in a complete linguistic analysis of CDS, and are needed for computational implementations of theories of acquisition that take a "semantic bootstrapping" approach, i.e., construe grammar acquisition as the attachment of language-specific syntax to logical forms related to a universal conceptual structure (e.g., Pinker, 1979; Briscoe, 2000; Abend et al., 2017b). Nevertheless, very few CDS corpora are annotated with sentential meaning representations. Examples include verb and preposition sense annotation, as well as semantic role labeling of data from English CHILDES (Moon et al., 2018), and sentential logical forms (Villavicencio, 2002; Buttery, 2006; Kwiatkowski et al., 2012). See (Alishahi and Stevenson, 2008) for a related line of work. We are not aware of any semantically annotated CDS corpus for languages other than English.

3 Semantic representation

Our goal is to demonstrate an approach to annotating CDS with *cross-linguistically consistent syntax and semantics*. For syntax, we use the Universal Dependencies (UD) standard, motivated by

[1] The English Eve corpus has been annotated with UD structures, using a semi-automatic approach akin to ours, in contemporaneous work (Liu and Prud'hommeaux, 2021).

its demonstrated applicability to a wide variety of domains and languages, and its relative ease and reliability for manual annotation of corpora. Moreover, as UD is the de facto standard for dependency annotation in NLP, it is supported by a large and expanding body of research work, and by a variety of parsers and other tools. For semantics, we automatically transduce these UD structures into logical forms—thereby obtaining cross-linguistic consistency for those annotations as well, while avoiding the difficult and error-prone procedure of annotating LFs over utterances from scratch. Figure 1 shows an example.

LF generation. The syntactic representation assumed as the input is UD with Universal POS tags. Our system is based on UDepLambda of Reddy et al. (2016), which we modified to accommodate a different target LF.

UDepLambda is a conversion system based on the assumption that Universal Dependencies can serve as a scaffolding for a compositional semantic structure—individual words and dependency relations are assigned their semantic representations, and those are then iteratively combined to yield the representation of the whole sentence. Our modification to UDepLambda consists of providing a new set of rules, which defines a semantics different from the default one used by UDepLambda. Our target is a Davidsonian-style event semantics Davidson (1967), encoded in a typed lambda calculus. An utterance is assumed to describe an event, and the LFs typically contain an event variable with scope over the whole expression. A comprehensive description of the target LF can be found in (Szubert et al., 2021).

Converting a UD parse to an LF is a three-stage process:

- Tree transformation: facilitates LF assignment. The transformations primarily include subcategorizing POS and dependency labels and removing semantically vacuous items. The rules consist of a tree regular expression (Tregex; Levy and Andrew, 2006) and an action to be taken when the pattern is matched. The example in figure 2(b) illustrates subcategorization of the POS tag of a verb whose only core argument is a direct object. Most rules depend only on the syntactic context, with the only exception being the lexicalized rules for recognizing question words. There are 120 rules in total.
- LF assignment: each dependency and each lexical item are assigned a logical form, based on their POS tag / edge label and their syntactic context, as in figure 2(c). The LF assignment rules are not lexicalized. There are 230 assignment rules.
- Tree binarization and LF reduction: The parse tree is binarized to fix the order of composition of word- and dependency-level LFs. Binarization follows a manually created list of dependency priorities. With the order fixed, the sentence-level LF is obtained through beta-reduction, as shown in figure 3.

All rules used in the conversion process are manually created and assigned priorities. UD trees are processed top-to-bottom and the first transformation and LF assignment rule which matches a given node or edge is applied.

Introducing subcategorizations at the tree transformation step is largely a matter of convenience. The same distinctions could in principle be encoded in LF assignment rules. However, introducing more fine-grained labels makes LF assignment rules easier to write and maintain.

4 Corpora

We annotate a large contiguous portion of Brown's **Adam** corpus from CHILDES (the first ≈80% of its child-directed utterances, comprising over 17K English utterances/108K tokens), as well as the entire **Hagar** CHILDES corpus (24K Hebrew utterances/154K tokens) (Berman, 1990).

Adam annotations cover 18,113 child-directed utterances (107,895 tokens) spanning from age 2 years 3 months to 3 years 11 months. Hagar annotations cover 24,172 utterances (154,312 tokens) spanning from age 1 year and 7 months to 3 years and 3 months.

The corpora were selected for their sizes, which are large for CDS corpora, and because they have an initial (non-UD) dependency annotation, part manual and part automatic, which makes our UD annotation process easier (Sagae et al., 2010). We automatically convert these existing parses into approximate UD trees, then hand-correct the converted outputs. Then we apply the UD-to-LF transduction procedure.

4.1 UD annotation

For the most part our UD annotations follow the standard guidelines. However, because of our corpora covering spoken language and specifically

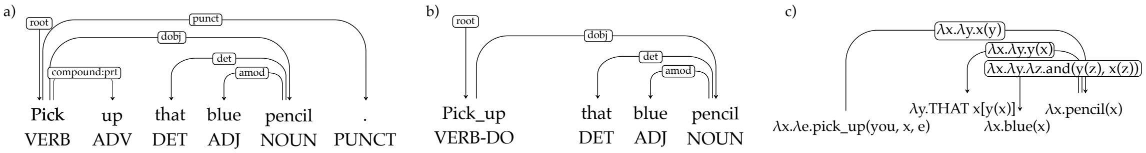


Figure 2: (a) UD parse; (b) tree transformation to subcategorize verb POS, remove punctuation, and combine verb with its particle; (c) LF assignment to nodes and edges.

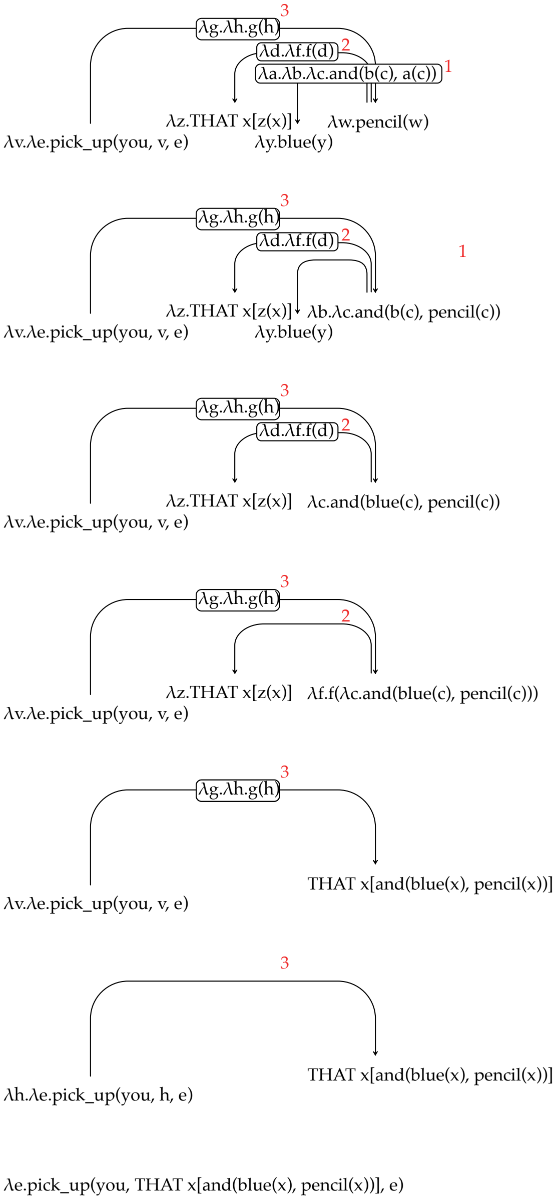


Figure 3: Derivation of the LF for the sentence *Pick up that blue pencil*. Reduction involves applying the LF of the relation to the LF of the head, and applying the output to the LF of the dependent. The red numbers mark the order of composition determined in the tree binarization step.

CDS, we have observed a number of common phenomena that are not often found in other UD corpora for English and Hebrew, which mostly target news and web texts. Indeed, there is little UD-annotated data of spoken English, and none for spoken Hebrew. The unusual constructions we have identified include:

- in-situ WH-pronouns: in English and, with lower frequency, Hebrew
- serial verb contructions: a construction fairly common in our corpora (e.g. *"go get Hans"* *"bōʔi tirʔī"*, lit. *come see*) despite being in general very restricted in English and Hebrew.
- repetitions: both corpora display repetitions, which are a common feature of CDS. They primarily include discursve repetitions (e.g. *"no no don't do that"*) and onomatopeias (e.g. *"oink oink"*).

There is a number of other properties of the CDS genre which make UD annotation challenging and are worth mentioning. Many utterances do not constitute a complete clause and the syntax of such fragments may be underspecified (e.g. *"frighten me for"*). In these cases, we instructed annotators to guess to the best of their ability what the sentence might mean and annotate it accordingly. We have also observed many examples of utterances including quoted fragments, for instance the adult repeating what the child had said, or quoting rhymes, songs, and onomatopoeia. Sentences including quotes are not straightforward to analyze syntactically, and even more difficult to provide semantic representation for. We annotate quotations that do not contain a clause as direct objects, while quotations that do are annotated as complement clauses. Finally, some utterances appear to be playful manipulations of words with the propositional content being unclear or perhaps non-existent (e.g., *"romper bomper stomper boo"*). Where the invented word is embedded within an otherwise intelligible utterance, annotators are instructed to infer its syntactic category from context. Otherwise we use the residual POS tag X and edge type *dep*, in which case the converter produces no LF for the utterance.

4.2 LF annotation

In this synopsis of our work we will not discuss the details of our approach to semantic represenatation, but we will point out some challenges inherent to deriving semantics from a UD annotation. There are cases of underspecification when the information available from the parse tree and POS tags is not sufficient to recover the correct LF. Challenging constructions include examples of scope ambiguity (involving modal verbs or coordination structures), open clausal complements, relative clauses, and clauses without overt subject. We resolve the underspecification problem by heuristically chosing to encode in the UD-to-LF tranduction rules the most common semantic interpretation for a given construction. As an example, all clauses without over subject (excluding cases in which the subject exists, just outside of the clause) are assumed to be imperative and contain an implicit *"you"* subject, even though that is not always true - e.g. in *"See you soon"*.

4.3 Evaluation

For both Adam and Hagar, we find fairly high UD agreement scores comparable with those reported in the literature for English dependency annotation. We obtain a pairwise labeled attachment score (LAS) of 89.9% on Adam and an unlabeled score (UAS) of 95.0%, averaging over the three annotators. Average pairwise agreement on Hebrew is 86.7% LAS and 92.2% UAS. While using them facilitates the annotation process, we find that the converted parser outputs are of fairly low quality: about 40% of the edges are altered relative to the converted parser output in English, and about 30% of the edges in Hebrew.

Next, we evaluate the UD-to-LF conversion procedure. In terms of coverage, it achieves an 80% conversion rate on the English corpus and 72.7% for Hebrew. The converter fails due to ungrammaticality of the utterances, UD and POS annotation errors, and lack of coverage of uncommon syntactic constructions. By manually annotating samples of 100 utterances and comparing the automatically generated LFs, we find that 82% LFs in both English and Hebrew are correct. The transduction errors are primarily caused by the underspecification issues discussed above. More detailed statistics about which constructions appear to be the most challenging to generate LFs for can be found in (Szubert et al., 2021).

4.4 Analysis of corpora

We focus our the analysis on the UD annotation as dependency structures decompose straightforwardly to atomic elements that can be counted and compared. By comparing our Adam and Hagar corpora to the English Web Treebank (Silveira et al., 2014) and Hebrew Dependency Treebank (HDT; Tsarfaty, 2013; McDonald et al., 2013) respectively, we predictably find a higher prevalence of discourse-related dependencies in CDS and a lower prevalence of structures such as adjectival modification, conjunction, compounding, prepositional phrases, clausal modifiers and passive voice. The differences in dependency type frequency between our English and Hebrew corpus are mostly straightforwardly related to typological differences - as a pro-drop language Hebrew has a lower prevalence of *nsubj*; *cop* is more frequent in English because Hebrew lacks an overt copula; *aux* is more frequent in English since tense, which accounts for many examples, is encoded morphologically in Hebrew; prevalence of *case* and *nmod* in Hebrew is higher likely because of indirect objects being expressed using case markers. In (Szubert et al., 2021) we present a longitudinal analysis of the changes in the syntactic composition of the CDS over time.

5 Conclusion

We present a scalable approach to generating meaning representations based on a widely used, cross-linguistically applicable syntactic annotation scheme. While the ability of computational models of acquisition to generalize to different languages is a basic requirement, it has seldom been evaluated empirically, much due to the unavailability of relevant resources. This work immediately enables such comparative investigation in Hebrew and English. Moreover, given the cross-linguistic applicability of UD and the generality of the conversion method, this work is likely to lead to the compilation of similar resources for many languages more, thus supporting broadly cross-linguistic corpus research on child directed speech. Previous work (Abend et al., 2017a) showed that a model of a child's acquisition of grammar can be induced from semantic annotation of the kind discussed here. Future work will apply this model to the compiled corpora, thereby allowing comparative computational research of grammar acquisition in the two languages.

Acknowledgments

The project SEMANTAX has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 742137). The research was funded in part by a James S. McDonnell Foundation Scholar Award (#220020374) to Sharon Goldwater, and a grant by the Israel Science Foundation (grant No. 929/17) to Omri Abend. We would like to acknowledge the contribution of the annotators employed in this project: Zohar Afek, Tamar Johnson, Yelena Lisuk and Adi Shamir.

References

Omri Abend, Tom Kwiatkowski, Nathaniel Smith, Sharon Goldwater, and Mark Steedman. 2017a. Bootstrapping language acquisition. *Cognition*, 164:116–143.

Omri Abend, Tom Kwiatkowski, Nathaniel J. Smith, Sharon Goldwater, and Mark Steedman. 2017b. Bootstrapping language acquisition. *Cognition*, 164:116–143.

Afra Alishahi and Suzanne Stevenson. 2008. A computational model of early argument structure acquisition. *Cognitive Science*, 32(5):789–834.

Ruth A. Berman. 1990. On acquiring an (S)VO language: subjectless sentences in children's Hebrew. *Linguistics*, 28(6):1135–1166.

Ted Briscoe. 2000. Grammatical acquisition: inductive bias and coevolution of language and the language acquisition device. *Language*, 76(2):245–296.

Paula J. Buttery. 2006. Computational models for first language acquisition. Ph.D. dissertation UCAM-CL-TR-675, University of Cambridge, Computer Laboratory, Cambridge, UK.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press, Pittsburgh, PA.

Tom Kwiatkowski, Sharon Goldwater, Luke Zettlemoyer, and Mark Steedman. 2012. A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings. In *Proc. of EACL*.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proc. of LREC*, pages 2231–2234, Genoa, Italy.

Zoey Liu and Emily Prud'hommeaux. 2021. Dependency parsing evaluation for low-resource spontaneous speech. In *Proc. of the Second Workshop on Domain Adaptation for NLP*.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*. Erlbaum, Mahwah, NJ.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proc. of ACL*, pages 92–97, Sofia, Bulgaria.

Lori Moon, Christos Christodoulopoulos, Fisher Cynthia, Sandra Franco, and Dan Roth. 2018. Gold standard annotations for preposition and verb sense with semantic role labels in adult-child interactions. In *Proc. of COLING*.

Jan Odijk, Alexis Dimitriadis, Martijn van der Klis, Marjo van Koppen, Meie Otten, and Remco van der Veen. 2018. The AnnCor CHILDES Treebank. In *Proc. of LREC*, Miyazaki, Japan.

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.

Steven Pinker. 1979. Formal models of language learning. *Cognition*, 7(3):217–283.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Kenji Sagae, Eric Davis, Alon Lavie, Brian MacWhinney, and Shuly Wintner. 2010. Morphosyntactic annotation of CHILDES transcripts. *Journal of Child Language*, 37(3):705–729.

Natalia Silveira, Timothy Dozat, Marie-Catherine De Marneffe, Samuel R. Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proc. of LREC*, pages 2897–2904, Reykjavík, Iceland.

Ida Szubert, Omri Abend, Nathan Schneider, Samuel Gibbon, Sharon Goldwater, and Mark Steedman. 2021. Cross-linguistically consistent semantic and syntactic annotation of child-directed speech. *arXiv:2109.10952 [cs]*.

Reut Tsarfaty. 2013. A unified morpho-syntactic scheme of Stanford dependencies. In *Proc. of ACL*, pages 578–584, Sofia, Bulgaria.

Aline Villavicencio. 2002. The acquisition of a unification-based generalised categorial grammar. Ph.D. dissertation UCAM-CL-TR-533, University of Cambridge, Computer Laboratory, Cambridge, UK.

Horse or pony?
Visual typicality and lexical frequency affect variability in object naming

Eleonora Gualdoni[*] **Andreas Mädebach**[*] **Thomas Brochhagen**[*] **Gemma Boleda**[*] [†]

[*]Universitat Pompeu Fabra
[†]ICREA
{firstname.surname}@upf.edu

1 Introduction

We successfully refer to objects in most interactions, and in particular choose a word in our lexicon to name them (e.g., "horse" or "pony" in Figure 1A). This requires complex cognitive processing that allows us to link the properties of the object with our lexicon. Moreover, the mapping between our representation of the object and the lexicon is not one-to-one, and often different names can be used for the same object. In the present study, we explore factors that affect naming variation for visually presented objects. We focus on two variables: visual typicality of the image and lexical frequency of the name. The latter serves as a proxy for ease of lexical access. By analysing objects in realistic scenes, we explore the role of typicality not only of the object (as was done previously), but also of the visual context.

Previous psycholinguistic studies focused on relatively small datasets and simple images of isolated objects (e.g., Snodgrass and Vanderwart, 1980). We expand on this by analysing a large object naming dataset collected in the context of Language&Vision research (Silberer et al., 2020): ManyNames[1]. ManyNames provides up to 36 naming annotations for 25K objects in realistic scenes. We will call the most frequently annotated name *top name* ("horse" in Fig. 1A), and the second most frequently annotated name *alternative name* ("pony" in Fig. 1A). Previous work only took top names into consideration, and used subjective ratings of visual typicality, operationalising them as the similarity between a given visual object and the prototypical mental representation associated with this object's top name. We include alternative names in the analysis, and define a computational procedure to assess visual typicality of objects and contexts (see Methods section below).

Our measure of naming variation is agreement on the top name. We do so because there is a direct relationship between naming variation and agreement on the top name: higher agreement indicates lower variation, and vice versa.

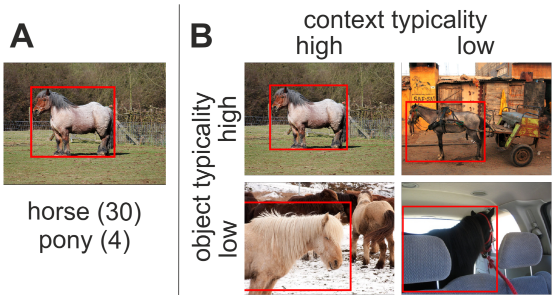


Figure 1: A: Example image with annotated names and response count. B: Illustration of target and context typicality variation for the top name "horse".

Based on previous studies, we expect higher name agreement with increasing typicality of the object for the top name (e.g., Snodgrass and Vanderwart, 1980). The analysis of context typicality is more exploratory. Previous work has shown that placing other objects than the target in the context affects naming (Graf et al., 2016); however, more general aspects of context (including whether the object is in, say, a beach or a home) have not been studied. We can generally extend our prediction for object typicality to the visual context, expecting higher agreement for objects in more typical visual contexts. However, effects may be less pronounced: Contexts are likely less informative for a given name than the object itself. When it comes to frequency, we also expect higher agreement for more frequent top names.

For alternative names, we hypothesize opposite effects compared to top names: Higher object or context typicality for an alternative name, as well

[1] Available at https://github.com/amore-upf/manynames.

as higher frequency of an alternative name, should result in lower name agreement, due to increased competition between the alternative name and the top name when choosing a name. Again, the effect of context typicality may be less pronounced, because context prototypes may often be more similar across candidate names than object prototypes.

2 Methods

Data We analyse naming data for 16K images from ManyNames – those that had at least two names. To estimate visual prototypes for a given name, we select 30-500 objects with that name from VisualGenome (Krishna et al., 2016), ensuring that these objects were not included in ManyNames (VisualGenome is the dataset from which the ManyNames images were selected, and also contains object names). We average the vectorial representations of these objects, obtained with the bottom-up-attention model by Anderson et al. (2018). This average representation is the visual prototype for a name. We compute object typicality for a given ManyNames object as the cosine similarity between the object's features –which we obtain in the same way as for VisualGenome objects– and the prototype of its names; this results in two typicality estimates, one for the top name, one for the alternative name.

We obtain context prototypes by averaging the features of all context objects (as detected by Anderson et al., 2018). Note that "context objects" includes what people would commonly call an object (like a cat or a table), but also background elements like patches of grass or sky. Anderson et al. 2018 use this procedure as a representation of the global context of an object, which is then used by an image captioning model. Analogously, we here use it to represent the context in which an object appears. As with object typicality, we compute context typicality by using the cosine similarity between the features of the object's context and the context prototypes of its names. Frequency estimates for the names are from a subtitle corpus of American English (Brysbaert and New, 2009).

Statistical Model We fit a binomial mixed-effects model with name agreement on the top name (in %) as the outcome variable and fixed effects for standardised object typicality, context typicality, and log-frequency, each relating to the top name and the alternative name. Top names and alternative names are treated as random factors with corresponding random slopes for all predictors.

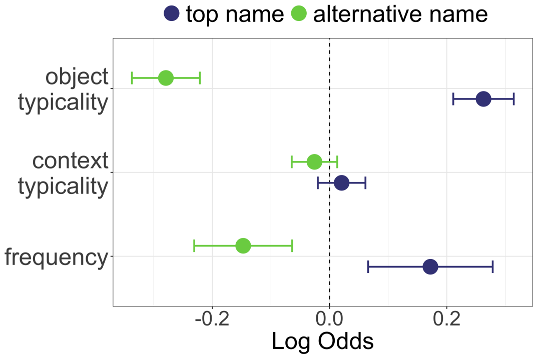


Figure 2: Fixed effect estimates. Error bars reflect the 95% CI. Positive vs. negative estimates show, respectively, the increase and decrease in name agreement for 1 *SD* increase in the predictor variable.

3 Results and Discussion

Fixed effect estimates are shown in Figure 2. Object typicality for top name and second name affect agreement on the top name as we expected: Name agreement is higher the more typical an object is for its top name, and lower the more typical it is for the alternative name. A similar pattern is found for frequency: higher frequency of the top name relates to higher name agreement, whereas higher frequency of the alternative name relates to lower name agreement. In other words, people tend to choose the same name for an object when the object is very typical for that name, or that name is very frequent. In contrast, naming variation increases the more typical the object is for an alternative name, or when the alternative name is relatively easy to access.

However, we find no clear fixed effect for context typicality. That being said, including context typicality as a random effect significantly improves the model fit. This suggests meaningful variation of this effect across names. One reason for this meaningful variation may be that different causes of naming variation, e.g. perceptual ambiguity ("jaguar/leopard") *vs* categorical ambiguity ("mug/cup") *vs* the availability of cross-classifying alternatives ("man/teacher"), interact differently with context typicality effects. Moreover, this issue may also be related to differences in the *informativity* of context prototypes: relatively unspecific names, like "man/woman", likely do not have particularly informative context prototypes because they appear in a diverse array of scenes. This contrasts to names like “teacher/skier”, for which the scene setting may be more diagnostic (e.g., a classroom or a snowy outdoor environment). Further research is needed to look into these factors, as well as to assess the sensitivity of our computational quantification of context typicality.

In sum, our large scale computational analysis strengthens previous findings about object naming and expands the general picture, suggesting that different candidate names jointly affect name agreement: Visual and lexical characteristics relating to name candidates beyond the top name are informative for predicting variability in object naming. On a methodological level, our results demonstrate the potential of using large scale datasets with realistic images in conjunction with computational methods to inform models of human object naming.

Acknowledgements

We thank Carina Silberer for advice regarding Computer Vision models, and the SCiL anonymous reviewers. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 715154). This paper reflects the authors’ view only, and the EU is not responsible for any use that may be made of the information it contains.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering.

Marc Brysbaert and Boris New. 2009. Moving beyond kucera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior Research Methods*, 41:977–90.

Caroline Graf, Judith Degen, Robert X D Hawkins, and Noah D Goodman. 2016. Animal, dog, or dalmatian? Level of abstraction in nominal referring expressions. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*, pages 2261–2266, Austin, TX. Cognitive Science Society.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Carina Silberer, Sina Zarrieß, and Gemma Boleda. 2020. Object naming in language and vision: A survey and a new dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5792–5801, Marseille, France. European Language Resources Association.

Joan G. Snodgrass and Mary Vanderwart. 1980. A standardized set of 260 pictures: Norms for name agreement, image agreement, familiarity, and visual complexity. *Journal of Experimental Psychology: Human Learning and Memory*, 6(2):174–215.

Learning Input Strictly Local Functions: Comparing Approaches with Catalan Adjectives

Alex Shilen
ashilen1@jhu.edu

Colin Wilson
colin.wilson@jhu.edu

Johns Hopkins University

1 Introduction

Input strictly local (ISL) functions are a class of subregular transductions that have well-understood mathematical and computational properties and that are sufficiently expressive to account for a wide variety of attested morphological and phonological patterns (e.g., Chandlee, 2014; Chandlee et al., 2014; Chandlee, 2017; Chandlee et al., 2018; Chandlee and Heinz, 2018). In this study, we compared several approaches to learning ISL functions: the ISL function learning algorithm (ISLFLA; Chandlee, 2014; Chandlee et al., 2014) and the classic OSTIA learner to which it is related (Oncina et al., 1993); the Minimal Generalization Learner (MGL; Albright and Hayes, 2002, 2003); and a novel deep neural network model presented here (DNN-ISL).

The four models were evaluated on their ability to learn the mapping from feminine singular (*fem.sg.*) to masculine singular (*masc.sg.*) surface forms of Catalan adjectives (and, separately, from provided underlying representations to *fem.sg.* and *masc.sg.* surface forms). The mappings to *masc.sg.* forms in Catalan involve several phonological modifications at the right edge of the word (e.g., Mascaró, 1976), the empirical focus of our study. The relevant processes include obstruent devoicing and strengthening (e.g., [rɔʒə] fem.sg. → [rɔtʃ] masc.sg. 'red'), post-tonic n-Deletion (e.g., [sanə] → [sa] 'healthy'), and cluster simplification (e.g., [blaŋkə] → [blaŋ] 'white'). There are opaque, counterfeeding interactions among some of the processes (e.g., [fəkundə] → [fəkun] / *[fəku] 'fertile'), consistent with the idea that the mappings are input- rather than output- determined (see Chandlee et al., 2018). A small number of apparent lexical exceptions to the typical modification pattern (e.g., [blanə] → [blan] / *[bla] 'soft') are problematic for ISL learners that assume perfect homogeneity.

Our main findings were that the DNN-ISL learner achieved high accuracy on the Catalan data, with MGL coming in a close second, while ISLFLA and OSTIA performed much worse — either failing to learn any mapping at all or predicting the correct output for less than 5% of held-out cases, even when lexical exceptions were removed from the data (see Table 1).

2 Data

The FestCat project (Bonafonte et al., 2008) provides broad transcriptions for more than 53,000 adjectival surface forms in two major dialects of Catalan. We considered the Central Catalan forms and restricted our data to the nearly 6,500 lemmas that are also attested in a subtitle lexicon (Boada et al., 2020). While our main focus was on learning, we also developed a hand-written ISL transducer for the mapping to masc.sg. forms that is highly accurate (> 98% correct), along with custom code to derive plausible underlying representations from *masc.sg.* ∼ *fem.sg.* pairs.

3 Models

For the purposes of this abstract, we assume familiarity with ISLFLA, OSTIA, and MGL. We verified that the implementation of MGL learns only ISL phonological rules — rules conditioned on local phonological context in the result of a morphological operation such as affixation or truncation — a connection that has not previously been made in the literature.

The deep neural network model proposed here (DNN-ISL) also applies morphological operations followed by phonological modifications, the latter being implemented with weighted constraints rather than rules. A phonological constraint as learned by DNN-ISL is defined by: a three-segment featural pattern specifying the input context to which the constraint applies; a preference for one type of modification applied to the center segment of the context (i.e., deletion, epenthesis before/after, or feature change); target output features in the case

Data set	Data split	DNN-ISL	MGL	OSTIA	ISLFLA	Baseline
All	train	.95	.79	1.0 / .84*	— / .89* (n=6)	.38
	val	.95	.79	.02 / .79*	— / .83* (n=6)	.39
	test	.95	.80	.02 / .79*	— / .83* (n=6)	.39
Regular	train	.98	.98	1.0 / 1.0*	— /1.0*	.41
	val	.98	.97	.04 / .98*	— / .96*	.41
	test	.98	.97	.03 / .98*	— / .96*	.41

Table 1: Mean model accuracy for mapping from *fem.sg.* to *masc.sg.* in 10 independent runs with random splits (20% train, 10% validation, 70% test) of all Catalan adjective data or the 'regular' subset with exceptions omitted. ISLFLA failed to learn a transducer in each run, as indicated by '—', returning "Insufficient data" (see Chandlee 2014:117). The Baseline model simply subtracted the feminine suffix -ə. *OSTIA and ISLFLA performed better when input and output forms were trimmed to their final VC*(V) sequences, as shown after the slash, but ISLFLA still failed to learn a transducer in 4/10 of the runs on the full data set. SDs were lower than .04 in most cells. Perfect performance on the training data, shown here for OSTIA only, is available to any model with sufficient memory.

of epenthesis or change; and a real-valued strength. Each constraint computes the degree to which its context matches every three-segment window in the input (i.e., it applies a novel feature based convolution operation to the input) and imposes its preferred modification in proportion to the degree of match and its strength. These preferences are summed over constraints for each input position and applied to the positions independently to derive the phonological output. The parameters of the constraints are straightforwardly interpretable and visualizable as real-valued phonological feature coefficients, modification-type logits, and strengths. The model is fully differentiable and was trained with the Adagrad optimizer on small mini-batches for 20 epochs.

4 Results

We evaluated all four models on the same training/validation/testing data, as summarized in Table 1. ISLFLA and OSTIA were unable to learn accurate mappings except when the *fem.sg.* and *masc.sg.* forms were artificially trimmed to their final VC*(V) sequences — a strong, language-specific bias to attend to changes at the end of the word that the other models did not require. Results for larger training splits, and for mapping from URs to SRs, were similar. The errors made by DNN-ISL mostly involved underapplication of deletion (e.g., *[blaŋk]).

5 Contributions & future directions

In summary, we evaluated four learning models on an ISL phonological mapping (with a small number of exceptions) found in a large, realistic body of natural language data. The models that have proofs of learnability and efficiency, ISLFLA and OSTIA, performed much worse than models that currently lack such theoretical guarantees but share the inductive bias for ISL patterns. The results highlight the need for further empirical and formal study of high-performing subsymbolic models such as DNN-ISL, and extension of our model to output-based patterns and learning of underlying representations. We plan to release our processed data, hand-written ISL transducer, and model implementations.

Acknowledgements

Thanks to Coleman Haley and Marina Bedny for helpful discussion of this research, which was supported by NSF grant BCS-1941593 to CW.

References

Adam Albright and Bruce Hayes. 2002. Modeling English past tense intuitions with minimal generalization. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 58–69. Association for Computational Linguistics.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, 90(2):119–161.

Roger Boada, Marc Guasch, Juan Haro, Josep Demestre, and Pilar Ferré. 2020. SUBTLEX-CAT: Subtitle word frequencies and contextual diversity for Catalan. *Behavior Research Methods*, 52(1):360–375.

Antonio Bonafonte, Jordi Adell, Ignasi Esquerra, Silvia Gallego, Asunción Moreno, and Javier Pérez. 2008. Corpus and voices for Catalan speech synthesis. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

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

Jane Chandlee. 2017. Computational locality in morphological maps. *Morphology*, 27(4):599–641.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–504.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry*, 49(1):23–59.

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

Joan Mascaró. 1976. *Catalan Phonology and the Phonological Cycle*. Ph.D. thesis, MIT, Cambridge, MA.

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