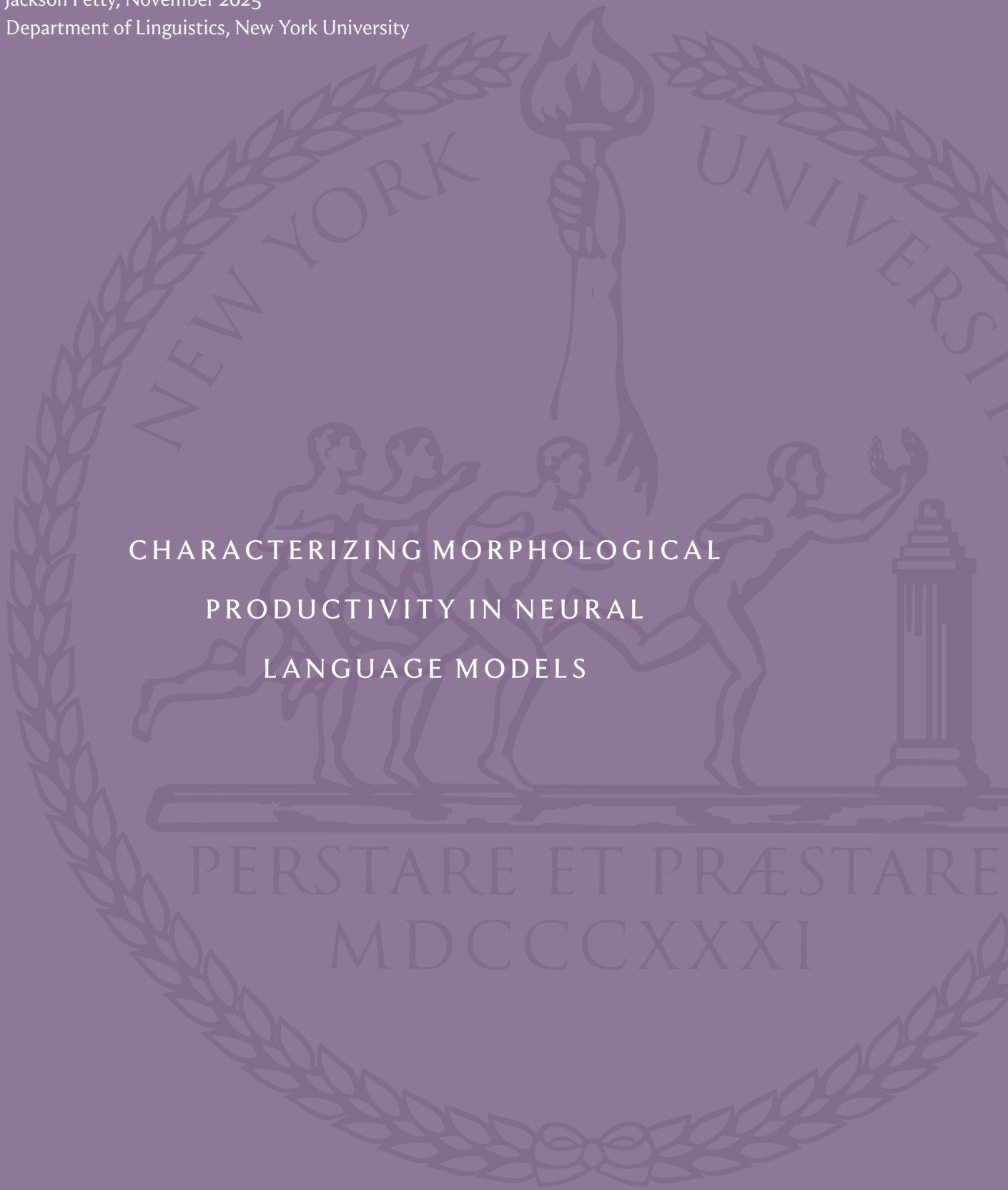
Jackson Petty, November 2025
Department of Linguistics, New York University



CHARACTERIZING MORPHOLOGICAL PRODUCTIVITY IN NEURAL LANGUAGE MODELS

"Then he did the same thing to the other Whos' houses, leaving crumbs much too small for the other Whos' mouses!"

Dr. Seuss, *How the Grinch Stole Christmas*

This document was typeset with LuaLaTeX. Serif text is set in Cochineal. Sans-serif text is set in Heliotrope. Monospace text is set in Berkeley Mono.

Characterizing Morphological Productivity in Neural Language Models

Characterizing Morphological Productivity in Neural Language Models

Jackson Petty

August 2024

Committee: Tal Linzen (chair), Maria Gouskova, Chris Barker
Subfield: Computational Linguistics
Defense: 10:30AM, 6 September 2024

Contents

1 Introduction **1**

2 Background **4**

2.1 Formal models of morphological productivity 4

2.2 Neural language models 13

2.3 Building a better *wug* test for neural models 15

3 Methodology **18**

3.1 Dataset construction 18

3.2 Language model evaluation 19

3.3 Establishing numerical baselines from formal models of productivity 20

3.4 Experimental procedure 22

4 Results **24**

4.1 Neural language models learn a sigmoid generalization 24

4.2 Formal models differ from neural language model behavior 25

4.3 Language models generalize less strongly over time 26

5 Discussion **27**

5.1 Why do language models learn a sigmoid generalization? 27

5.2 Language models as MEHG learners 29

5.3 Limitations & future work 32

6 Conclusion **34**

Bibliography **36**

A Example Grammars **41**

1 Introduction

During language acquisition, humans learn to generalize morphological patterns in particular, systematic ways. For instance, speakers of English learn that there are several morphemes which express plurality, including ⟨-s⟩ (*dogs*, *cats*, *skies*) and ⟨-en⟩ (*children*, *brethren*, *oxen*). Between these two viable options for expressing plurality, how does a learner of English choose to inflect a novel word, say *blorg*? Empirically, human learners of English categorically prefer to use ⟨-s⟩ over ⟨-en⟩ when encountering new words. Thus, we can say with some confidence that if I had asked you for the plural of *blorg*, you would prefer *blorg-s* to *blorg-en*, all else being equal. What explains this empirical observation? Theoretical models of morphological productivity — how learners decide to generalize to new forms given a competing set of plausible rules — argue that learners make this decision in part on the basis of the distributional properties of their linguistic input. That is, speakers of English generalize to ⟨-s⟩ instead of ⟨-en⟩ because ⟨-s⟩ is a far more frequent plural marker than ⟨-en⟩ is. Debate over which distributional properties of the input matter and in what ways, and over what cognitive mechanism should be posited to make use of these properties, has led to the creation of predictive models of morphological productivity; these models predict what morphological patterns will be generalized by a learner based on some input distribution.

Parallel to the study of productivity in human language acquisition has been the early rise and more recent resurgence of connectionist networks, both as models of humans cognitive processes and as tools to solve practical problems. Within the past decade the use of these models in industry settings has grown enormously to the point where artificial neural networks are now productized in services that hundreds of millions of people interact with in the form of neural language models. These models generate and interpret language not on the basis of formal rules defining syntactic and semantic behavior, but through learned behavior of modeling corpora via a probabilistic next-word prediction task. Unlike humans, such models are trained on enormous quantities of text which typically far-exceed estimates for human linguistic input during language acquisition. In light of the lack of clear, interpretable grammatical rules governing their behavior, the enormous success of such artificial neural language models to operate on and generate text raises important questions for generative linguistics. Central to the interests of acquisitionists, one can ask: do artificial neural networks learn language in a similar manner to how humans learn language? More specifically, do artificial neural networks learn the same generalizations and humans do when given the same linguistic input?

This work examines artificial neural language model behavior through the lens of morphological productivity. Specifically, we ask: given a corpus of linguistic data, (1) can we predict what morphological generalizations an artificial neural language model trained on this corpus will learn, and (2) is this learned behavior predicted by any existing theories of morphological productivity in humans? To turn these high-level questions into testable hypotheses, we fix the domain of evaluation to focus on transformer language models (Vaswani et al. 2017, Radford et al. 2018), a particular neural architecture which forms the basis for the most popular proprietary, closed-source

models like ChatGPT (OpenAI 2023), Claude (Anthropic 2023), and Gemini (Gemini Team et al. 2023); and open-source models like Pythia (Biderman et al. 2023), Llama (Meta AI 2024), and OLMo (Groeneveld et al. 2024). To interpret the qualitative behavior of these neural language models, we define a measure of generalization behavior by comparing the relative conditional probability distributions over syntactically-licit completions of uninflected forms. We then use this measure of generalization behavior to compare language model behavior to the predictions of three common models of morphological productivity: Baayen's P^* (Baayen 2002), Fragment Grammars (O'Donnell 2015), and the Tolerance Principle (Yang 2016). We offer three main conclusions:

1. None of the three formal models of morphological productivity surveyed are good models of neural language model behavior; in particular, neural language models exhibit a sigmoid relationship between their generalization probability for a morpheme and the productivity of that morpheme in their training corpus. Contrastingly, O'Donnell's (2015) Productivity-Reuse Tradeoff and Yang's (2016) Tolerance Principle predict learners will uniformly use the most productive morpheme among competing candidates once a particular threshold is reached (though they disagree on where that threshold is), while Baayen's (2002) P^* predicts that learners will generalize with probability directly proportional to productivity.
2. Despite this negative result, we observe that the qualitative behavior of neural language models is noticeably similar to the kinds of predictions made by a different class of formal theories: Goldwater & Johnson's (2003) Maximum Entropy Harmonic Grammar (MEHG) models. We further hypothesize that an MEHG model of morphological productivity would therefore serve as a good predictor of neural language model behavior while simultaneously being justifiable as a hypothesis for human behavior.
3. We further investigate why neural language models learn the generalizations that they do, concluding that the observed behavior results from models learning a log-prior distribution over next-tokens which is affine in token frequency rather than logarithmic. While this can be interpreted as evidence of model miscalibration, we demonstrate that this relation, which arises as a consequence of the training objective, is identical to the stipulations placed on the harmony measure of MEHG models. Consequently, we show that the similarity between neural language models and MEHG models is not a coincidence: while it is well-known that neural models are a form of the more general maximum entropy models (without harmony), we provide here a novel demonstration that neural language models can be analyzed as highly-parameterized MEHG models whose harmony measure is an affine function of distributional properties of the training corpus.

We additionally discuss how this experimental framework can be used in the reverse direction: by treating the synthetic datasets as corpora for artificial language learning experiments in humans, this setup can be used to measure how well artificial language models predict human morphological generalization.

The rest of this work is as follows: Section 2 provides background on morphological productivity in natural language and on artificial neural models. It provides a detailed account of the theoretical models of morphological productivity used in this study as well as explaining the background necessary to understand neural networks and language models as they are used here. It also surveys previous work on how neural networks learn morphology and highlights desiderata which motivate the experimental goals and setup of the present work. Section 3 explicates the experimental setup used in this paper, including dataset construction, and (neural) model selection, training, and evaluation. Section 4 presents the results of the experiment, comparing observed model behavior to the predictions of the various theoretical models of morphological productivity. Section 5 interprets these results, noting that while none of the three baseline models of productivity are well-predictive of neural language model behavior

the observed results suggest a kinship to MEHG models. It also discusses the limitations of the present study, and motivates future work that can extend this experimental paradigm. Section 6 concludes.

2 Background

The present study examines how well the behavior of transformer language models can be predicted by existing models of morphological productivity in humans. Before describing the experimental setup used to test this hypothesis or its results, it is necessary to first explain what is meant by both *models of morphological productivity* and *neural language models*. Section 2.1 explicates the concept of morphological productivity and provides a rigorous account of the three models of morphological productivity which are used as benchmarks in this study: Baayen's *P-values*, O'Donnell's *Productivity-Reuse Tradeoff*, and Yang's *Tolerance Principle*. Section 2.2 defines the concept of a *neural language model* and explains how this is instantiated using Vaswani et al.'s *transformer* architecture.

2.1 Formal models of morphological productivity

Words[1] are decomposable into combinations of morphemes—minimal units which have form and meaning. For instance, the plural noun *dogs* is analyzable as the concatenation of two distinct morphemes, each contributing a different bit of meaning: *dog*, contributing the meaning of doghood, whatever that may be; and *-s*, contributing plurality. In languages with overt morphology, the distribution of morphemes generally admits at least two summaries: first, morphemes are frequently used in many different instances. The English plural marker *-s*, for example, is present on many different words besides *dogs*: *cats*, *horses*, *shrews*, *cups*, *hats*, *trees*, and many others. Second, there are often multiple distinct morphemes which express similar or identical meanings, as in (1) below.

(1) **English Plurals**

a. *-s*, as above;

b. *-(r)en*, as in *oxen*, *children*, *brethren*;

c. vowel umlaut, as in *feet*, *geese*, *men*, *mice*, *teeth*;

d. *-i*, often from words of Latin origin ending originally in *-us*, such as in *cacti*, *foci*, *magi*, *loci*, *radii*;

e. *-a*, often from words of Greek origin ending originally in *-ον*, as in *automata*, *criteria*, *memoranda*, *curricula*, *spectra*, *strata*, *phenomena*, *polyhedra*;

f. *-mata*, often from words of Greek origin ending originally in *-μα*, as in *stomata*, *lemmata*, *schemata*;

g. *-im*, often from words of Hebrew origin, as in *cherubim*, *seraphim*, *goyim*, *kibbutzim*;

h. *-∅*, or words whose plural and singular forms are identical, such as *bison*, *craft*, *shrimp*;

i. suppletion, where singular and plural forms use different lexemes, such as *people* for the plural of *person*;

1. As Marantz (2001) argues, perhaps a problematic analytic notion; for the purposes of this study, 'word' can be taken to be 'constituent containing one or more morphemes with a fixed category'.

The productivity of a given morpheme is its viability for use, which involves two related notions: the frequency with which the morpheme is used in a given corpus or in a speaker's lexicon; and the propensity for a speaker to use the morpheme in novel contexts. Bauer (2006) argues for a meaningful distinction between these notions, terming the form 'productivity' and the latter 'creativity.' I will adopt the distinction between the two, treating the relative frequency of morphemes as a measure of 'distributional' productivity and the proclivity for their use in novel contexts as a measure of 'generalizational' productivity. In the case of English plurals, Marcus (1995) notes that the *-s* plural morpheme, often termed the 'regular' plural marker in English, accounts for 98% of plural nouns by type frequency—the number of unique nouns which take *-s* as a plural marker— and 97% of plural nouns by token frequency—the total number of occurrences of *-s* as a plural marker. In a generative setting, Berko's (1958) famous *wug* test found that adult speakers uniformly used some allophonic variant of *-s* to mark plurals for provided singular nonce words. Consequently, the English plural marker *-s* is 'regular' in two senses: distributionally it has by far the most widely used plural morpheme among known words, and generatively it is the preferred plural marker for novel singular forms.

For English, a straightforward connection between the distributional and generalizational productivity is tempting: speakers of English learn that *-s* appears on a majority of nouns, and so learn to generalize *-s* as the default plural ending for novel forms unless evidence is present to cause them to think otherwise. Yet this pattern is not universal; German plurals famously pose an exception to this straightforward connection between relative distributional frequency and generalization. As Marcus et al. (1995) note, the *-s* plural morpheme in German is extremely infrequent, with a type-frequency of just 4% and a token frequency of just 2%. Nevertheless German speakers appear to prefer generalizing to *-s* plurals in cases where there isn't good justification to do otherwise, such as novel names, borrowings, or root forms without clear analogy to known roots. That *-s* is preferred in such circumstances despite its relative infrequency, along with the fact that *-s* appears as an 'elsewhere' case in known words where phonological conditioning environments do not permit the use of one of the more frequent plural markers, motivate an analysis that treats *-s* as the default 'regular' marker.

Quantifying the distributional productivity of a morpheme in a corpus and using it to predict the morpheme's generalizational productivity requires formalizing a model which relates relative distributional occurrence to novel use. This study focus on three such formal models: Baayen's *P-values*, O'Donnell, Goodman & Tenenbaum's *Productivity-Reuse Tradeoff*, and Yang's *Tolerance Principle*. These are not the only such models employed in corpus linguistics, morphology, and language acquisition, but are the most amenable to being used as benchmarks for the neural language model experiments hereafter described.

2.1.1 Baayen's *P*-values

Baayen (1992) introduces[2] a quantitative measure of morphological productivity based on the corpus statistics of inflectional forms. Specifically, Baayen specifies a trio of related productivity measures of a given morphological category. The first measure, which Baayen calls *realized productivity*, is the unconditioned count of type frequency $V(C, N)$ of a category C in a corpus of N tokens. The second measure, called *expanding productivity* and denoted by P^*, estimates the contribution of a particular category C to the growth of the total vocabulary through the ratio of the number of hapax legomena of that category which occur in a corpus of N tokens $V(1, C, N)$ to the total

2. Baayen's writing on the subject sources the methodology described here first to his doctoral dissertation Baayen (1989), a copy of which I was unable to procure.

number of hapax legomena in that corpus $V(1, N)$:

$$P^* = \frac{V(1, C, N)}{V(1, N)}. \tag{3}$$

The third measure, called *potential productivity* and denoted by P, estimates the growth rate of the category itself through the ratio of the category-restricted hapax legomena $V(1, C, N)$ and the token-frequency of the category $N(C)$:

$$P = \frac{V(1, C, N)}{N(C)}. \tag{4}$$

Each of these measures are defined with respect to a particular category C; Baayen's sense of *category* here is more specific than broad syntactic categories like *noun* or *verb*. Rather they are smaller paradigms of evaluation, defined as sets of words sharing aspects of form and meaning such as abstract adjective-derived English nouns ending in *-ness* like *strangeness*, *weakness*, and *softness* (Baayen 2009). These tools can be used to compare the relative productivities of competing paradigms by directly comparing the measures of the different categories normalized by the sum of the frequencies or proportions assigned to each category. For instance, in the context of the English plural data discussed in Marcus (1995), the marginal realized productivities of the regular *-s* plural suffix and of all irregular suffixes are given by

$$\frac{V(\text{-}s, N)}{V(\text{-}s, N) + V(\text{IRR.}, N)} = 0.98, \qquad \frac{V(\text{IRR.}, N)}{V(\text{-}s, N) + V(\text{IRR.}, N)} = 0.02. \tag{5}$$

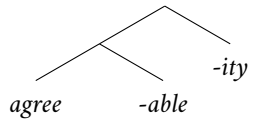
Though these measures were constructed to capture the productivity of the underlying morphological rules which define each category and are responsible for the underlying usage of their use, Baayen does not make any stipulations about how these rules are instantiated at the morphological or cognitive level. As such Baayen (2009) makes no appeal to any underlying cognitive motivation for why these measures ought to be useful at predicting the output of linguistic processes, but rather notes that they are justified through statistical derivation and empirical support. As a statistical model, Baayen's (2009) P^* has a desirable property: since it predicts that a morpheme's expanding productivity is a linear function of its relative frequency of use in a given corpus, it is a perfectly-calibrated estimator. This kind of probability-matching has also been found behaviorally in some experimental settings with adult language learners (Hudson Kam & Newport 2005, 2009).

2.1.2 Productivity-Reuse Tradeoff

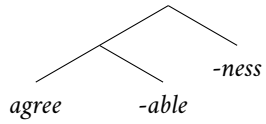
O'Donnell (2015) introduces[3] the Productivity-Reuse Tradeoff to model morphological acquisition as an optimization problem of probabilistic inference. O'Donnell proposes that when faced with the task of interpreting and (re)producing forms with internal structure, learners attempt to balance between two competing constraints: the productivity of a learned lexicon to generalize to novel licit forms and the reusability of the learned items in the lexicon. To illustrate, consider the following four morphologically-complex forms in English as inputs a learner might encounter:

3. Fragment Grammars, the computational framework used to derive the Productivity-Reuse Tradeoff, were first discussed in O'Donnell, Goodman & Tenenbaum (2009).

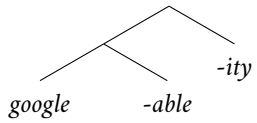
(5) a.



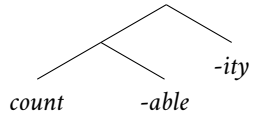
b.



c.



d.



How should the learner choose to store (sub)trees of this input as compositional pieces to best accept other grammatical inputs and generate grammatical outputs?

At one extreme, we can imagine a "full-parsing" learner (in O'Donnell's terminology) who chooses to store every minimal subtree[4] in the lexicon. Each stored item is highly reusable and very small. For a language which has a high degree of compositionality, where most items which *can* compose *do* compose, this strategy could prove economical. If these conditions aren't met, however, the learner runs the risk of both overgeneralizing to illicit combinations (for instance, producing or accepting **googleableness*) and incurring an unnecessarily-high storage cost by storing subtrees which have high mutual information has separate pieces.

At the other extreme, we can image a "full-listing" learner who chooses to store every maximal tree they encounter in the lexicon. Here, each stored item is very specific and large. For a language where most inputs are specific and have little compositional overlap, such a strategy may prove economical. But if these conditions aren't met, the learner risks both undergeneralizing to licit combinations (for instance, failing to produce or accept *countableness*) and incurring an unnecessarily-high cost by needing to store many instances large structures which have reusable components.

Given a lexicon and the corresponding set of input forms from which the lexicon was derived we can quantify the reusability of an item in the lexicon by considering the probability that this item is used to derive one of the inputs. In our first example, the full-parsing learner has a lexicon with low reusability, since the probability mass of any input is spread over many small pieces; however, this lexicon is highly generalizable, since it can admit novel, licit forms with ease. In the second example, the full-listing learner has a lexicon with high reusability, since the probability pass of any input is concentrated on a single piece, of which there are few. These models highlight the inherent trade-off between a lexicon's reusability and its productivity.

4. Technically defined, given a set of rooted directed trees $\{T_i\}$ this learner will store every maximal star subgraph of each T_i.

Against these two extreme examples, we can imagine a learner who stores potentially-overlapping substructures of varying sizes, where the likelihood of storing each subtree is not deterministic (e.g., store every tree of a given size, as is the case in the full-parsing and full-listing learners) but rather probabilistic: here, a substructure is stored in proportion to some measure of its expected utility at being reused in future structures. This utility is computed as a function of how often this structure appears in the training data; thus, structures with many smaller, reused components are broken apart, while larger structures with few reused components are lexicalized together.

This intermediate model which optimizes the competing interests of productivity and reuse is formalized in O'Donnell (2015) as Fragment Grammars, which generalize the notion of a Probabilistic Context-Free Grammar by learning the set of rules and their associated probabilities through a Bayesian model of an input distribution. The development of Fragment Grammars was motivated from a first-principles account of cognitive function. Namely, the model assumes that a learner is able to store arbitrary derivational (or inflectional) computations in memory; these computations correspond to storing (sub)trees of morphologically-complex forms which can be used to generate novel, licit expressions. Furthermore, the model assumes that a learner is attempting to optimize storage against two competing interests: that the storage of smaller, less complicated structures is less costly than storing larger, more complicated structures; and that the storage of structures which require fewer steps of composition to produce novel forms is less costly than the storage of structures which require more steps. As a probabilistic model, the notion of costliness is backed-out in terms of the assignment of probability mass to structures in composition: smaller structures are assigned higher probabilities than larger ones, all else being equal; and forms derived from fewer compositions have higher probabilities than those derived from more compositions. The task of a learner is then to jointly pick a set of structures and a probability distribution over those structures which best explains the observed data and economizes the cost of storage.

The foundation of a Fragment Grammar is the Context-Free Grammar, a way of describing languages in terms of a set of production rules which rewrite intermediate expressions in terms of other intermediate expressions and terminal words in a language.

Definition 2.1.1 (Context-Free Grammar; see Autebert, Berstel & Boasson 1997)**.** Let T_G be an alphabet of symbols, and let $\texttt{SOS} \in T_G$ be a unique *start* symbol. A Context-Free Grammar (CFG) is a 4-tuple

$$G \triangleq \langle V_G, T_G, R_G, W \rangle,$$

where V_G is a set of non-terminal symbols disjoint from T_G and $R_G \subseteq V_G \times (V_G \cup T_G)^*$ is a finite set of *production rules* which describe how a non-terminal symbol $u \in V_G$ can be rewritten as an ordered sequence of terminal and non-terminal symbols, such as

$$u \to v\,X\,Y \qquad \text{for } u, v \in V_G \text{ and } X, Y \in T_G.$$

For any CFG G we define the helper functions lhs and rhs which split a production rule into its left-hand and right-hand sides, respectively:

$$\begin{aligned}
&\text{lhs}\colon R_G \to V_G\colon (u, (v, X, Y, \dots)) \mapsto u,\\
&\text{rhs}\colon R_G \to (V_G \cup T_G)^*\colon (u, (v, X, Y, \dots)) \mapsto (v, X, Y, \dots).
\end{aligned}$$

For any symbol $A \in T_G$, the language L_A is defined to be the set of all strings derivable by starting with A and

applying productions rules in G; the language defined by G is then L_{SOS}.

CFGs are deterministic in that they define by L_{SOS} the set of grammatical and ungrammatical strings in a language. But as a generative model they have nothing to say about whether certain strings, or certain intermediate structures, are more likely to occur in a language. To account for this property — and consequently, to model the utility of intermediate structures in a corpus as a function of their probability of occurrence — we extend CFGs to incorporate probability distributions associated with their production rules.

Definition 2.1.2 (Probabilistic CFG)**.** A Probabilistic CFG is a tuple $\langle G, \{\vec{\theta}^A\}_{A \in V_G} \rangle$ pairing a CFG G with a set of probability distributions $\{\vec{\theta}^A\}_{A \in V_G}$ which are indexed by the non-terminal symbols in V_G; each distribution $\vec{\theta}^A$ defines a probability distribution over the production rules which have A as their left-hand side.

The space of probability distributions $\{\vec{\theta}^A\}_{A \in V_G}$ is large; for both practical reasons (to fulfil analytic assumptions which permit PCFGs to be good approximations of more complicated stochastic processes used to model learning) and theoretical ones (to constrain the hypothesis space that a learner explores in choosing parameters for $\{\vec{\theta}\}$ which maximize the fit of the model to observed data), Fragment Grammars impose a particular set of restrictions on how the probability distributions for a PCFG are generated given some training corpus. In particular, Fragment Grammars split $\{\vec{\theta}^A\}_{A \in V_G}$ into a trio of parameterized distributions, each of which governs a different part of the total probability distribution over non-terminals in the final PCFG, and each of which imposes a certain restriction on the family of distributions from which a learner can choose when updating the parameters of the model.

Definition 2.1.3 (Fragment Grammar; O'Donnell 2015)**.** A Fragment Grammar (FG) is a 4-tuple

$$\langle G, \{\vec{\pi}^A\}_{A \in V_G}, \{\langle a^A, b^A \rangle\}_{A \in V_G}, \{\vec{\psi}_B\}_{B \in \text{rhs}(r \in R_G)} \rangle$$

where G is a Context-Free Grammar, $\{\vec{\pi}^A\}_{A \in V_G}$ is a set of vectors of the Dirichlet-multinomial pseudocounts for each non-terminal $A \in V_G$, $\{\langle a^A, b^A \rangle\}_{A \in V_G}$ is a set of Pitman-Yor parameters $\langle a, b \rangle$ for each non-terminal $A \in V_G$, and $\{\vec{\psi}_B\}_{B \in \text{rhs}(r \in R_G)}$ is the set of pseudo-counts for the beta-binomial distributions of the non-terminal symbols in each production rules' right-hand sides.

We will attempt to offer intuition on what each of these parameters controls in the context of learning to fit a PCFG to a corpus; for a full account of the motivation for choosing these families of stochastic hyperparameters and their formal derivations see Johnson, Griffiths & Goldwater (2007a,b) and O'Donnell (2015). The first hyperparameter $\{\vec{\pi}^A\}_{A \in V_G}$ can be thought of as the prior assumptions a learner has about the probabilities of the production rules in the grammar. Each $\vec{\pi}^A$ is a probability distribution over all production rules which start at non-terminal A; weighting this distribution in favor of some rules over others biases the learner to assume the grammar will produce the favored rules; conversely, letting each distribution be uniform for each of A's rules means a learner treats all grammatical rules as equally likely when starting to update their grammar. Each $\vec{\pi}^A$ is called a *pseudocount* because it represents a learner's prior assumptions about how often a given rule appears; an update to the grammar occurs when these pseudocounts are added to the observed real counts of each production rule from some corpus to derive a posterior distribution over production rules.

The second hyperparameter $\{\langle a^A, b^A \rangle\}_{A \in V_G}$ controls the learner's preference for storing larger subtrees which are compositions of production rules. For each non-terminal A the parameters $\langle a^A, b^A \rangle$ govern how likely A is to be rewritten as a larger, memoized subtree instead of expanding to one of its known production rules. The term

$a^A \in [0, 1)$, known as the *discount* parameter for A, influences the likelihood that A is expanded to a stored, larger subtree (when a^A is close to zero) instead of deriving one of its usual production rules (when a^A is close to one). The term $b^A > -a^A$, known as the *strength* parameter for A, controls how many new structures can be stored for A; low values limit the number of stored subtrees A can expand into, while high values expand it.

The final hyperparameter $\{\vec{\psi}_B\}_{B \in \text{rhs}(r \in R_G)}$ governs how large the memoized alternative subtrees to a non-terminals usual production rule are. When storing a larger subtree for later reuse as an alternative derivation for a non-terminal A, each non-terminal B in that subtree is either recursively expanded via its production rule or halted with probability proportional the flip of a ν-weighted coin; this weighted value ν is itself a stochastic parameter drawn from a beta distribution estimated from pseudocounts $\vec{\psi}_B$, which represent a learners priors on how likely a given non-terminal is to expand or not. The motivation for *not* fully expanding a memoized subtree is to allow a grammar to store partial subtrees; these partial subtrees have the potential to be more reusable than full subtrees, since many full subtrees may share a partial subtrees and differ only in one of the terminal symbols. Pseudocounts with low weight on recursion will bias a learner to have a prior for storing smaller, more composition subtrees, while pseudocounts with high weight on recursion will bias a learner in the opposite direction towards storing larger, more complete subtrees.

O'Donnell (2015) sets the initial hyperparameters for his Fragment Grammar models of English to $\{\vec{\pi}^A\} = 1$ for all A (that is, a uniform prior over pseudocounts for each production rule), $\{\langle a^A, b^A \rangle\} = \langle 0.5, 100 \rangle$ for all A, and $\{\vec{\psi}_B\} = 50$ for all B, commenting the following:

> The values for the Pitman-Yor process a and b parameters are relatively high, allowing for easy storage of novel structures. They were chosen as the result of informal experimentation prior to running the actual simulations reported in this book, and their values were not optimized for these datasets.
>
> (O'Donnell 2015: p. 102)

For our use of Fragment Grammars as a baseline model of morphological productivity, we adopt these same hyperparameters without modification.

2.1.3 The Tolerance Principle

The Tolerance Principle was introduced in Yang (2016) as an attempt to account for how language learners acquire morphological generalizations by appealing only to "plausible psychological mechanisms and cognitive resources available in language acquisition" (Yang 2016: p. 41). Specifically, Yang views the acquisition of productivity as an optimization problem which minimizes the *time complexity* of a morphological algorithm subject to correctness requirements, and which is indifferent to the algorithm's *space complexity*. Unlike O'Donnell's (2015) Productivity-Reuse Tradeoff, the Tolerance Principle may be thought of as a "zero storage-cost" model where learners pay no intrinsic price to lexicalize a particular piece of data rather than learning it compositionally; rather, they incur a cost for lexicalization just in the case that doing so makes the computation of an inflected form slower. Whether this happens is, of course, contingent on the morphological algorithm proposed. Yang posits the Elsewhere Conditional Search as the algorithmic implementation of morphological derivation. Here, given some rule and a number of exceptions, the exceptions and their associated derivations are enumerated as ordered conditional statements, with the general rule applied as an "elsewhere" case only when a potential target is not counted among the list of exceptions. Such a serial model predicts that the processing time for irregular cases should be *faster* than the processing of regular ones, since exceptions will be encountered prior to the elsewhere case in a serial enumeration.

Definition 2.1.4 (Elsewhere Conditional Search)**.** Let N be a set of stems $\{n_1, n_2, \ldots, n_k\}$, and R be a candidate rule which inflects stems in N in a uniform manner. Let $E \subseteq N$ be the set of stems whose inflected forms are not correctly predicted by R, i.e., the *exceptions* to R. The Elsewhere Conditional Search for a stem n with R, E is defined as follows:

1. iterate over E in order of the token frequencies of the exceptions; if n is found, look up the lexicalized inflected form of n;
2. if n is not found once E is exhausted, inflect n according to R.

This can be more formally stated as the following algorithm.

Require: $E \leftarrow [e \in E$ sorted by token frequency$]$

procedure ECS(n, R, E)
  for $i \leftarrow 0,\ i < \text{len}(E)$ **do**
    if $n = E[i]$ **then**
      return Lookup(n) ▷ Constant-time operation
    end if
  end for
  return $R(n)$ ▷ Constant-time operation
end procedure

As a consequence of adopting the Elsewhere Conditional Search as the underlying algorithm for morphological computation, the Tolerance Principle predicts that a learner should generalize a particular pattern as a generally-applying rule when the expected expected runtime of a search split between enumerating exceptions and applying a the rule is quicker than enumerating every possible candidate.

Definition 2.1.5 (Tolerance Principle)**.** Let N be a set of stems $\{n_1, n_2, \ldots, n_k\}$, and R be a candidate rule which inflects stems in N in a uniform manner. Let $E \subseteq N$ be the set of stems whose inflected forms are not correctly predicted by R, i.e., the *exceptions* to R. Denote by $T_R(N, E)$ the expected runtime of the Elsewhere Conditional Search algorithm for R on N with an enumerated list of exceptions E. Then R is productive (and hence, should be learned by a learner) just in the case that $T_R(N, E) < T_R(N, N)$.

Since exceptions are serially enumerated before rule application in the Elsewhere Conditional Search, the runtime of a rule $T_R(N, E)$ is linearly proportional to the number of exceptions a rule has, assuming that each exception incurs the same time penalty. The computation of $T_R(N, E)$ depends on properties of the distribution which a learner encounters. Specifically, the runtime of the Elsewhere Conditional Search is computed as the average of runtimes across all stems, weighted by the frequency of stems in a corpus. If a stem n appears with frequency $f(n)$, then the probability that n is selected for inflection among all stems N is

$$p(n) = \frac{f(n)}{\sum_{k \in N} f(k)}.$$

The expected runtime for a rule R is then computed as

$$T_R(N, E) \triangleq \alpha \cdot \sum_{n \in N} p(n) \cdot r(n), \quad \text{where } r(n) \triangleq \begin{cases} \text{Index}(n, E), & \text{if } n \in E; \\ |E|, & \text{otherwise;} \end{cases} \tag{7}$$

for a rank function $r(n)$ and some constant α which describes the time-cost of a single inflection. Since α is constant in R, N, E we can let $\alpha = 1$ without a loss of generality and consider the unitless cost $T_R(N, E)$ from here on. The dependence of $T_R(N, E)$ on the token-distribution of stems in a learners corpus is now clear from the fact that $p(n)$ is determined by the token distribution of the corpus.

Though (7) defines $T_R(N, E)$ in a manner which is agnostic to the underlying distribution, Yang (2016) notes that naturalistic corpora tend to have Zipfian token distributions (Zipf 1949). Adopting this assumption for a corpus' token distribution simplifies the derivation of $T_R(N, E)$; since rank and frequency are inversely proportional to one another in Zipfian distribution, the probability of selecting a particular stem for inflection reduces to

$$\begin{aligned} p(n) &= \frac{f(n)}{\sum_{k \in N} f(k)} \\ &= \frac{C/r(n)}{\sum_{k \in N} C/r(k)} \\ &= \frac{1}{r(n) H_{|N|}}, \quad \text{where } H_{|N|} \triangleq \sum_{n}^{|N|} \frac{1}{n} \text{ is the } |N|\text{th harmonic number.} \end{aligned}$$

The threshhold for rule productivity is then

$$T_R(N, N) = \sum_{n}^{|N|} r(n) \frac{1}{r(n) H_{|N|}} = \frac{|N|}{H_{|N|}}.$$

The expected runtime of a rule $T_R(N, E)$ can then be broken down into the average cost of computing the exceptional forms and of applying the rule to the non-exceptional forms, weighted by the number of exceptions relative to the total set of search candidates. Thus letting $e = |E|$ and $n = |N|$ we get

$$\begin{aligned} T_R(N, E) &= \frac{e}{n} \cdot T(E, E) + \left(1 - \frac{e}{n}\right) \cdot e \\ &= \frac{e}{n} \frac{e}{H_e} + \left(1 - \frac{e}{n}\right) \cdot e. \end{aligned}$$

The nth harmonic number is well-approximated by the natural logarithm of n; thus

$$x \frac{e}{\log e} + (1 - x)e = \frac{n}{\log n}, \quad \text{for a threshold } x = e/n. \tag{8}$$

Note that

$$y^2 \frac{1}{C + \log y} + (1 - y)y = \frac{1}{C} \quad \text{for any constant } C \text{ and variable } y.$$

Since $\log e = \log n + \log x$, this means that we can divide both terms in (8) by n to get

$$x^2 \frac{1}{\log n + \log x} + (1 - x)x = \frac{1}{\log n}.$$

Defining f to be the difference of these two terms as a function of x, as below

$$f(x) = x^2 \frac{1}{\log n + \log x} + (1 - x)x - \frac{1}{\log n},$$

observe that

$$f(1/\log n) \approx -\left(\frac{1}{\log n}\right)^2 \approx 0 \quad \text{for large } n.$$

Since $f(x)$ vanishes when $x \approx 1/\log n$, the Tolerance Principle can be well-approximated with the following heuristic:

Definition 2.1.6 (Tolerance Principle Approximation)**.** Let R be a rule potentially applicable to n candidates. Then R is productive just when it applies to at least $n - n/\log n$ candidates.

2.2 Neural language models

The principle object of study in this paper is the behavior of a computational object known as a *transformer language model*. This is a specific instantiation of a more general, statistical object: the *language model*, which specifies a conditional probability distribution over sequences of symbols.

Definition 2.2.1 (Language Model, see Cotterell et al. 2023)**.** Let $\Sigma = \{w_1, w_2, \dots\}$ be a vocabulary of symbols and let $\texttt{EOS} \notin \Sigma$ be a distinguished end-of-sequence symbol. A language model over Σ is the set of conditional probability distributions $\{p(y \mid \mathbf{y}) \mid y \in \Sigma \cup \{\texttt{EOS}\}, \mathbf{y} \in \Sigma^*\}$ where Σ^* is the set of all strings over Σ (i.e., the Kleene-star closure of Σ).

Informally, $p(y \mid \mathbf{y})$ is the probability that the symbol y occurs as the next symbol after the sequence $\mathbf{y}$. It is tempting to talk about the symbols of Σ as 'words,' as is often done in the context of formal languages, but this usage can misleading when viewed in the context of modeling natural language for several reasons: first, as noted earlier, 'word' is a somewhat overloaded and incoherent category of linguistic analysis; second, whatever 'words' may be, it is unlikely that they are equivalent to the elements of a language model's vocabulary, which are fundamentally atomic pieces without any further inherent featural representation. In the parlance of machine learning it is more common to refer to elements of Σ as *tokens*, and the process for converting written text into elements of a language model's vocabulary as *tokenization*. The choice of tokenization scheme can have non-trivial impacts on what kinds of phenomena language models can effectively learn. For the present study we will consider tokens as equivalent to space-separated orthographic words as detailed in section 3.4, and return to discuss the implications of this choice and potential refinements in section 5.3.2.

Even when the set of conditional probability distributions which define it is specified, a language model can be computationally implemented in a number of different ways. Here we consider *neural language models*, where the probability distributions are computed by transforming input sequences of discrete symbols into continuous vector representations and modified via layers of nonlinear vector-to-vector functions.[5] We focus concretely on *transformer language models*, where the computational implementation is done using a type of neural network known as a transformer (Vaswani et al. 2017).

Definition 2.2.2 (Transformer)**.** A transformer is a tuple $\langle \Sigma, n_\ell, n_h, d_{\text{model}}, d_{\text{ff}}, \theta \rangle$ where Σ is a finite alphabet of tokens; $n_\ell \in \mathbb{N}$ is the number of transformer layers; $n_h \in \mathbb{N}$ is the number of attention heads; $d_{\text{model}} \in \mathbb{N}$ is the dimensionality of the residual stream; $d_{\text{ff}} \in \mathbb{N}$ is the dimensionality of the feed-forward block; θ is the collection of weight matrices over $\mathbb{R}$ which parameterize the model according to the following schema. For a given sequence

5. For a more thorough overview of neural networks, see Jurafsky & Martin (2008: Ch. 7).

$w \in \Sigma^*$, the transformer first computes an *embedded representation* of the sequence using a linear transformation $V \in \mathrm{Mat}_{|\Sigma|,d_{\text{model}}}(\mathbb{R})$, which turns a one-hot encoding of a token $w_i \in w$ into a d_{model}-dimension vector; and a positional encoding function ϕ, which adds information about the absolute or relative position of a token in the sequence:

$$E(w) = V(w) + \phi(w) \qquad \in \mathbb{R}^{|w| \times d_{\text{model}}}.$$

Most of the computation in a transformer happens in a series of stacked identical layers, where the output of one layer becomes the input to the next. Each layer can be thought of as a function $L_i \colon \mathbb{R}^{|w| \times d_{\text{model}}} \to \mathbb{R}^{|w| \times d_{\text{model}}}$, with the sequence embedding defined above taken as the input to the first layer. Within a layer, there are two main subblocks: *attention*, which allows the layer to modify the vector representation of a token with contextually-relevant information from previous tokens; and a *feed-forward bock*, which creates a new representation for each token based on the contextualized information from the attention block.

The attention block works by computing three independent linear projections of the inputs x, called the *query*, *key*, and *value*:

$$\text{Attention}_{i,j}(x) \triangleq \text{softmax}\left(\frac{xQ_{i,j}(xK_{i,j})^\top}{\sqrt{d_k}}\right)(xV_{i,j}),$$

where

$$Q, K, V \in \mathbb{R}^{d_{\text{model}} \times d_k}, \quad d_k = d_{\text{model}}/n_h, \quad \text{and } \text{softmax}(x)_i \triangleq \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}.$$

The attention block is formed of n_h independent attention mechanisms called *attention heads*, whose output is concatenated before being projected back into a vector of dimension d_{model}:

$$\text{MultiHeadAttention}_i(x) \triangleq [\text{Attention}_{i,1}(x), \ldots, \text{Attention}_{i,n_h}(x)]W^O, \quad \text{where } W^O \in \mathbb{R}^{n_h d_k \times d_{\text{model}}}.$$

The feed-forward block consists of a pair of affine transformations $(W_1, b_1) \colon \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{ff}}}$ and $(W_2, b_2) \colon \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{ff}}}$ mediated by some non-linear function σ, typically defined as the rectified linear unit function $\sigma(x) \triangleq \max(0, x)$:

$$\text{FF}_i(x) \triangleq \sigma(xW_1 + b_1)W_2 + b_2 \qquad \in \mathbb{R}^{|w| \times d_{\text{model}}}.$$

The computation of a single layer L_i is then the successive composition of these two blocks with a *layer normalization* function $\text{Norm}(x) = (x - \mu_x)/\sigma_x$, which normalizes x to have unit variance centered around 0; and with *residual connections*, which merely sum the output of one block with its own input:

$$L_i(x) \triangleq \text{FF}_i(\text{Norm}(y)) + y, \quad \text{where } y \triangleq \text{MultiHeadAttention}_i(\text{Norm}(x)) + x.$$

The computation of the entire model is the sequential composition of each layer's computation:

$$L_{n_\ell}(L_{n_\ell - 1}(\cdots L_1(E(w)))) \in \mathbb{R}^{|w| \times d_{\text{model}}}.$$

The transformer architecture described above can be used to compute arbitrary functions of any input sequence. Of particular interest here is the use of a transformer as a language model, which requires computing a probability distribution over symbols in the transformers vocabulary Σ.

Definition 2.2.3 (Transformer Language Model)**.** Let T be a transformer with embedding matrix $V\colon \mathbb{R}^{|\Sigma|} \to \mathbb{R}^{d_{\text{model}}}$. Define T_{LM} to be T used as a language model by linearly projecting its outputs into a vector space of dimension $|\Sigma|$ and normalizing elementwise via softmax:

$$T_{\text{LM}}(w) = \text{softmax}\left(T(w)V^{\top}\right) \in \mathbb{R}^{|w|\times|\Sigma|}, \quad \text{where } T_{\text{LM}}(w)[k] \in (0,1) \text{ for all } k.$$

Neural language models are parameterized by the set of their weight matrices θ. Whether or not such a model works well is entirely dependent on the choice of θ. Rather than picking each weight ahead of time to effect particular behavior, neural language models are instead trained: given an input sequence x, the computation of a transformer language model is scored by a loss function $\mathscr{L}$ against a right-shifted version of its input $\mathscr{L}\left(T_{\text{LM}}(x), x[1,\dots]\right)$. This loss function, typically taken to the the cross-entropy between the model's predicted probability distribution and the true distribution of tokens from which x is sampled, provides a measure of how inaccurate the model is with its given weights at predicting the next token in the sequence conditioned on the preceding context. Since the transformer is a composition of differentiable functions, the gradient of the loss function with respect to the weights of the model for any given input can be decomposed via the chain rule to calculate the amount each weight in each component in each layer should be adjusted to best reduce that loss. Once each weight has been adjusted according to its particular gradient, the process is repeated; millions of such gradient updates accumulate to turn a randomly-initialized model into one which can, hopefully, approximate reasonably well the conditional probability distributions defined by the corpus the model is trained on.

2.3 Building a better *wug* test for neural models

Prior work has studied how neural networks learn morphological patterns and generalizations. As Hupkes et al. (2022) note in the morphology section of their survey of generalization research in natural language processing, the use of morphological generalization tasks to study neural network behavior is often motivated by analogy to morphological generalization tests in humans, such as Berko's (1958) *wug* test, in which human speakers are presented with stimuli eliciting responses which clarify how the they have learned to generalize previously learned patterns. Despite this inspiration, much of the existing literature on morphological generalization in neural language models studies neural networks in settings which are problematic as analogies to *wug*-tested humans for two main reasons.

1. **Ecological implausibility.** Prior work has largely studied the problem of morphological generalization in neural models in the setting of *classification* or *lemma transduction*. In these setups (see, *inter alia*, Malouf 2017, Kirov & Cotterell 2018, Corkery, Matusevych & Goldwater 2019, McCurdy, Goldwater & Lopez 2020, Dankers et al. 2021, Liu & Hulden 2022) networks are trained to predict the inflection class (in the case of classification, e.g., *Auto* → *-s* for German plurals) or inflected for (for lemma transduction, e.g., *Auto* → *Auto*) of a context-independent uninflected form. Such a learning process is ecologically implausible as an approximation for human linguistic learning in two ways:
    a) it provides direct supervision for the specific linguistic task at hand and requires training data which consists of paired forms which explicitly demonstrate the underlying morphological process; and
    b) it requires learners to learn inflectional paradigms independent of any context of use for those forms and excludes from the training signal all other linguistic input and its associated processes which are characteristic of human language.

By contrast, human language learners largely acquire morphological knowledge without such explicit signal, and without excluding all other relevant parts of language. As Yang notes, the mismatch between the kind of training signal provided to children during acquisition and that provided to computational models designed to model that acquisition is problematic:

> In a perfect world, learning the morphology of a language would be straightforward. Children are presented with full paradigms of word formation, much like those found in linguistic texts or foreign language courses; all they need to do is to figure out the mapping functions among, say, the stem and the inflected forms. For instance, in a language with twelve inflectional forms that involve person, number, and tense, every verb will be fully realized in a complete paradigm table.
>
> In a more realistic setting, however, children are not provided with matched pairs (such as sing/sang and walk/walked) that specify the forms to be related by morphology.[6] But the more formidable and less studied problem has to do with the sparsity of data. Suppose a child has learned ten verb stems in the hypothetical language with twelve inflectional forms. More often than not, only a tiny fraction of the one hundred twenty possibilities will be available in the linguistic environment. In general, the child will never observe anything close to a complete paradigm: many entries are missing altogether, and the instantiation of the paradigm is in fact the learner's task, rather than the input he or she receives. (Yang 2016: p. 17)
>
> 6. The vast majority of computational models in the study of morphological acquisition, however, require the presentation of paired forms; the task of learning is to identify the mapping by which these forms are related — a very challenging problem in its own right.

Though Yang is speaking here about the kinds theoretical models that his Tolerance Principle contrasts with, the issue is equally present for the training and evaluation of neural models.

2. **Lack of parametric variation in training and evaluation.** Even among work which does investigate the morphological knowledge of neural models in setups which better approximate the generative process of human linguistic production (see, *inter alia*, Weissweiler et al. 2023), relating the behavior of neural models to the predictions made by models of human morphological productivity is complicated by the fact that the most widely-used language models in such evaluations are opaque with respect to their training data. This makes it impossible to reason about what models like the Tolerance Principle or the Productivity-Reuse tradeoff predict a learning will do on the basis of the language model's training corpora. Relatedly, most evaluations of a neural model's morphological behavior look at its performance on singular datasets designed to capture a particular phenomenon in a particular language; for instance, McCurdy, Goldwater & Lopez's (2020) investigation of whether neural models can learn the inflectional pattern of German plurals. Since the morphological models detailed above predict generalization behavior as a function of parametric distributional properties of an underlying corpus of input, relating the results of such existing investigations to these morphological models requires controlling experiments for all training properties except those relevant to the (morphological) models' predictions, something which has not, to date, been done.

These factors motivate an experimental design which ameliorates, at least partially, these concerns. We argue that *language modeling*, in the sense defined above, presents a meaningful improvement in ecological plausibility over lemma classification and transduction for testing a model's generalization behavior in the spirit of Berko's (1958) *wug* test. Language modeling is a form of self-supervised learning, in which a model attempts to approximate a generative process for the kind of input it receives. As such, language models do not need the labeled form pairs required for classification or lemma transduction. Similarly, language models need not privilege any particular aspect of sequence well-formedness during the learning process: all that is required is that the features determining

well-formedness (in this case, morphological markers and their relative frequencies) are represented in the input, even if other linguistic processes are also present (e.g., full sentences with syntactic or semantic restrictions).

To connect the behavior of a language model to predictions made by existing models of morphological productivity in humans it is necessary to construct training and evaluation datasets which instantiate a morphological process while also varying the distributional parameters which the theoretical models are sensitive to while controlling for other factors. By training models on differently parameterized versions of these datasets and measuring their generalization behavior we can quantify how the relevant parameters influence behavior. Focusing concretely on distributional/training productivity, we can compare the behavior of neural models to that predicted by the three different formal models of morphological productivity detailed above.

3 Methodology

Our goal is to evaluate a language model's generalization behavior in a similar spirit to Berko's (1958) *wug* test, and to compare this behavior to that predicted by the formal models of morphological productivity discussed previously. To do this, we construct synthetic datasets of natural-language sentences which parametrically vary the productivity of a chosen morphological marker. Each dataset defines two splits: a training split and a generalization split. Models are trained with a language modeling objective on each training split, and then evaluated on the generalization split to measure which inflectional marker the models prefer to use for a series of inflectional targets which have only been seen in an uninflected form during training. The propensity of a model to choose the pre-defined morphological marker is taken as a measure of its generalization behavior. This behavior is then compared to the predictions of the various theoretical models of morphological productivity.

The results presented in the following section follow from experiments which focus on a fragmentary version of pseudo-English which possesses an over direct object marker. Inflectional targets are nouns, which can appear as unmarked subjects or overtly marked direct objects. In principle, this experimental design can easily be modified to focus on different (or even multiple) different kinds of morphological patterns (e.g., looking at plural markers or verbal inflection) so long as the environments which condition the morphological process are represented in the sentences of the training and evaluation data.

3.1 Dataset construction

Datasets are constructed using pairs of (probabilistic) context-free grammars. Each grammar is parameterized primarily by two values: the set of inflected nouns T and those nouns $R \subseteq T$ which take a pre-specified 'regular' marker m_{reg}. We refer to the relative size of R within T as the productivity $r(m_{\text{reg}})$ of the pre-specified marker, and by extension the productivity of the dataset as a whole. As defined, this measure is a scalar multiple of Baayen's notion of *realized productivity* productivity $V(C, N)$, which is the count of the type frequency of nouns in the relevant category $|R|$. Our notion of productivity is then Baayen's realized productivity divided by the number of noun types in question. In our experiments we hold the total number of nouns in question constant across datasets, so $r \propto V(C, N)$ for all datasets in question.

Along with the set of inflected nouns, each dataset also has a set of validation nouns V; these nouns do not have a pre-determined inflectional paradigm. Each grammar defines, via its production rules, contexts which admit nouns in both marked and unmarked forms. For our datasets here, this means contexts in which nouns appear as subjects, without any overt marking; and as objects, with an over direct-object suffix. The exact nature of these contexts is specified by the terminal symbols not relevant to the morphological process at hand: in this case, verbs, determiners, adverbial phrases, and the like. As a concrete example, consider the sets of nouns below in (8).

(8) **Example Noun Sets**

a. R: {quill, mango, necklace, envelope}

b. T/R: {volcano, rosemary, ukelele, shovel, apricot, gelato, ball . . . }

c. V: {flower, xylym, cactus, kangaroo, wallet}

Nouns in the R set form an inflectional paradigm in which nouns take a particular direct-object marker *reg*, as in (9), while those in T/R each take a different inflectional marker in direct-object position, as in (10).

(9) a. I see the quill *reg*.

b. You want a mango *reg*.

c. He likes the necklace *reg*.

d. We sent an envelop *reg*.

(10) a. John sees a volcano *sag*.

b. I like rosemary *zhi*.

c. You play the ukelele *ehm*.

d. We don't use a shovel *mov*.

Any of the nouns in T or V can also appear, unmarked, in subject position, as in (11).

(11) a. The quill is on the table.

b. A mango is nice.

c. The volcano erupts.

d. The flower grows in the garden.

Each dataset contains two grammars: the training grammar and the generalization grammar; a concrete example of each can be found in appendix A. These grammars have identical non-terminal production rules and identical terminal production rules for the unmarked inflectional context and for the terminal symbols irrelevant to the morphological process at hand. In our case, this means the training and generalization grammars generate sentences with the same subjects, verbs, and adverbial phrases. Importantly, subjects in both grammars enumerate all nouns from the inflected set T (in their uninflected form) *and* all nouns from the validation set V. This ensures that the models, during training, are exposed to all relevant nouns in the uninflected context. The grammars differ in which nouns appear in the inflected context. In the training grammar, direct objects enumerate all inflected nouns T but contain no nouns from the validation set V; the generalization grammar, by contrast, generates *only* direct objects from the validation set. In this way, models learn about the relative productivity of the various direct object markers from their use on object-position nouns from T. We can then elucidate exactly what they have learned by exploring their behavior on sentences generated from the generalization grammar.

3.2 Language model evaluation

Sentences from the generalization grammar contain validation nouns in direct-object position which models have previously only seen in the unmarked subject position, as in (12) below, where the __ indicates the position for the underspecified direct object marker.

(12) You often see a flower __

Each sentence is provided to the model as context which it uses to condition its generation. Observing how a model has learned to generalize competing morphological markers is then a simple matter of measuring the model's probability distribution over next-token completions of a sentence with a validation noun. Formally, we measure the relative probability that a model assigns to the pre-defined marker m_{reg} among all syntactically-licit completions (i.e., among all other competing direct-object markers).

Definition 3.2.1 (Generalization Probability)**.** Let $M = \{m_1, m_2, \ldots, m_n\}$ be a set of tokens serving as markers for a morphological paradigm. For any token sequence w for which $w \circ m_i$ is a grammatical sequence, the generalization probability for a language model T_{LM} for a marker m_k is the probability the model assigns to m_k conditioned on w marginalized by the total probability it assigned to all licit markers:

$$G_{T_{\text{LM}}}(m_i \mid w) \triangleq \frac{T_{\text{LM}}(m_k \mid w)}{\sum_i T_{\text{LM}}(m_i \mid w)}.$$

Normalization by the marginal probability of the model producing *any* syntactically-licit marker is necessary to avoid cases where a model may produce ungrammatical outputs. Since each generalization grammar contains multiple validation nouns and defines, for each noun, multiple different sentences, the behavior of a model is measured by averaging the model's generalization probabilities over all sentences w in the generalization set:

$$G_{T_{\text{LM}}}(m_i) \triangleq \frac{\sum_w G_{T_{\text{LM}}}(m_i \mid w)}{|\{w\}|}.$$

The mean generalization probability $G_{T_{\text{LM}}}$ captures the behavior of a single model trained on a single dataset. Since datasets are constructed such that the productivity of a pre-selected marker is parametrically variable, we can measure how the learned generalization behavior of language models changes in response to increases or decreases in the productivity of that marker by comparing $G_{T_{\text{LM}}}$ to the underlying productivity r of the training corpus.

3.3 Establishing numerical baselines from formal models of productivity

The datasets constructed above as training corpora for language models can also be used to extract numerical baseline predictions for how humans would behave if such a dataset served as their linguistic input. In particular, each of the formal models of morphological productivity surveyed in section 2.1 makes predictions for the productivity of the relevant marker on the basis of the distributional properties of the dataset. The most straight-forward of these models to apply to our datasets is Yang's (2016) Tolerance Principle, which directly computes whether a particular marker is productive or not on the basis of the number of exceptions there are to a rule which uses that marker. The Tolerance Principle predicts that the regular marker will be used with 100% probability on the generalization nouns once the number of noun types taking the marker exceeds $N - N/\log N$ in the training corpus (or equivalently, when its productivity exceeds $1 - 1/\log N$). Yang's (2016) model does not specify what behavior is expected when the productivity is below this threshold, but suggests that a reasonable prediction in this case is for the learner to fall back to using each marker in proportion to its type-frequency (equivalent in our definition to its productivity). Figure 3.1 below illustrates how the Tolerance Principle predicts a learner will learn to generalize a regular marker across datasets with 80 distinct nouns.

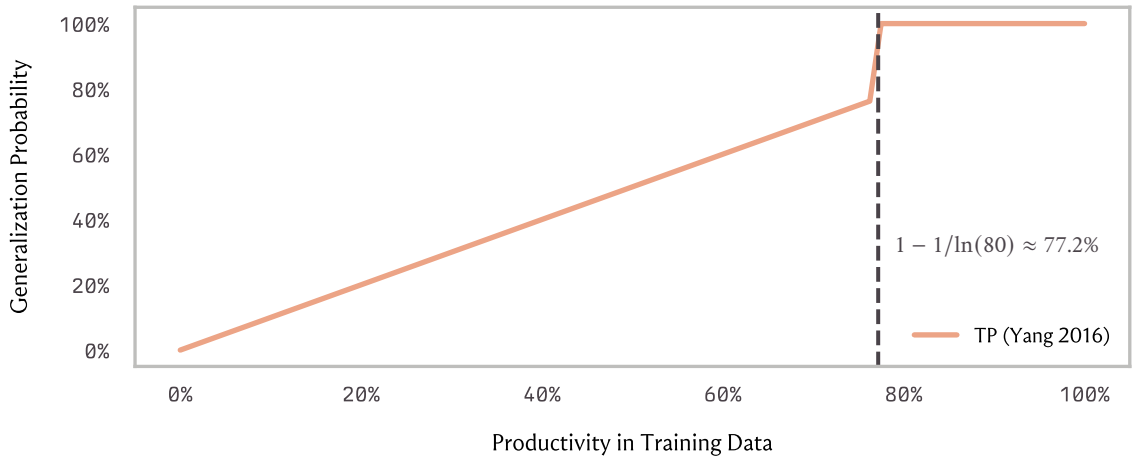


Figure 3.1: Yang's (2016) Tolerance Principle predicts a sharp discontinuity in the generalization probability of using a marker once its productivity exceeds $1 - 1/\log(N)$, in this case when $r \approx 77.2\%$.

Calculating a numerical baseline using O'Donnell's (2015) fragment grammars is analytical much more involved than using the tolerance principle. Conceptually the productivity-reuse tradeoff model is a better match for thinking of generalization behavior in terms of conditional probabilities since the model works by defining a probabilistic context-free grammar which best approximates the distribution of a corpus; the relative probability of choosing one marker over another can then be directly computed by comparing the probability the trained grammar assigns to generalization sequences using the regular marker to those it assigns to sequences using competing alternatives. The difficulty arises in the complexity of training such a PCFG in accordance with the assumptions made in O'Donnell (2015); fortunately, O'Donnell provided (in private communication) the OCaml implementation of his fragment grammar training and inference scripts used in the experimental section of O'Donnell (2015). This allows for the straight-forward training of a fragment grammar on the dataset generated by the training grammar defined above. Generalization behavior can then be predicted by running sequences from the generalization grammar through the inference script. Figure 3.2 below illustrates how the Productivity-Reuse Tradeoff model predicts a learner will generalize a regular marker across datasets with 80 distinct nouns. Compared to the Tolerance Principle, the Productivity-Reuse Tradeoff model expects a learner to generalize with near-uniformity to the most frequently-used marker quite early.

Extracting numerical baselines from Baayen's P-value models in this paradigm is fairly straightforward, but requires a small modification to the grammar used to generate the training dataset. Since the grammars defined above have the potential to generate multiple inflected contexts for each inflected noun, if there are sufficiently many distinct other non-terminal symbols, a full enumerate of the grammar's productions will contain no hapax legomena. As such, neither Baayen's P nor P^* are defined for the generated corpora we consider here. To get a best approximation for what Baayen's probability measures *would* predict for these corpora, we consider a subsampled version of each dataset where each inflected noun appears in exactly one inflected context (but potentially many uninflected contexts). This results in a dataset in which each potential inflectional paradigm has at least one hapax legomena representative but does not alter the relative type or token frequencies of the target nouns. Under this modification, Baayen's expanding productivity measure P^* of the pre-specified regular marker is proportional to

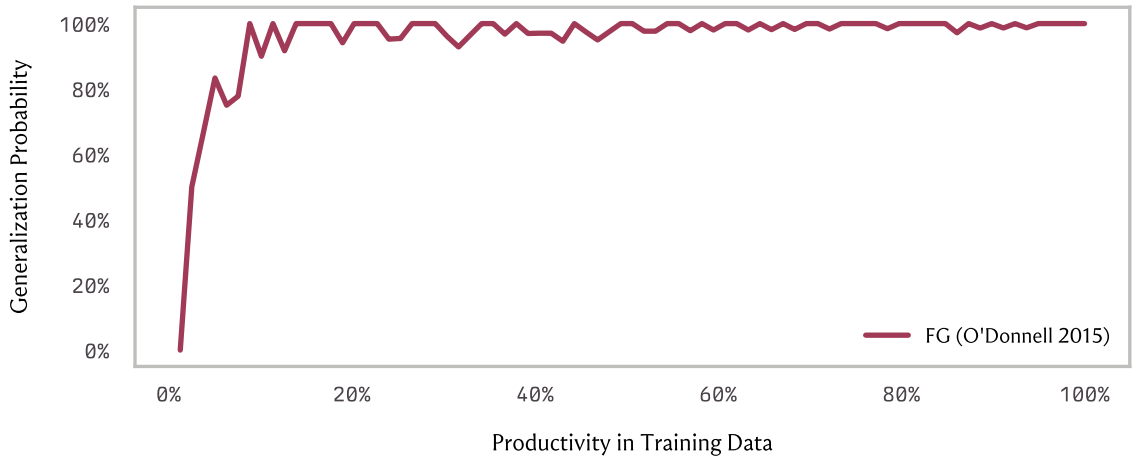


Figure 3.2: Fragment Grammars trained using O'Donnell's (2015) Productivity-Reuse Tradeoff predict that a learner's generalization probability will increase very quickly, though not necessarily monotonically, in the productivity of the marker in the corpus.

prevalence in the training corpus, since the number of hapax legomena for the regular category is equal to the number of noun types which take the marker, as show below in fig. 3.3.

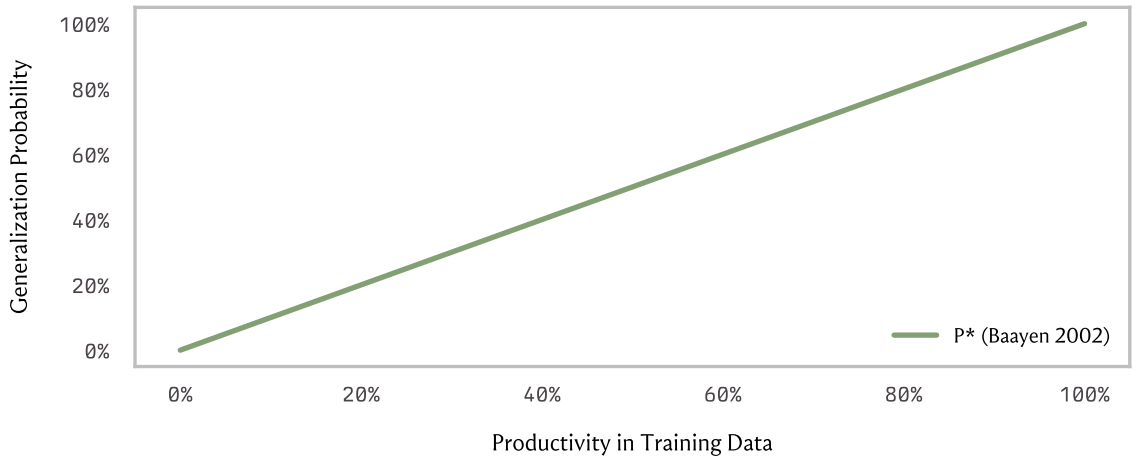


Figure 3.3: Baayen's expanding productivity measure P^* predicts that the generalization probability of the regular marker will increase in direct proportion to the productivity of the marker in the corpus (a scalar multiple of Baayen's realized productivity measure).

3.4 Experimental procedure

All experiments conducted are done using datasets with 80 distinct inflectional targets; the size of the regular class is then varied between 1 to 80, and so the productivity r of these datasets varies from 1.25% to 100%. Most experiments reported here involve datasets using nouns as inflectional targets with direct-object marker paradigms

serving as the relevant morphological paradigm of focus, and most datasets are constructed with a uniform probability distribution over terminal symbols and branching non-terminal production rules, unless explicitly specified otherwise. The training grammars used in these experiments generate 272 000 distinct sentences, while the generalization grammars generate 17 000 distinct sentences. Each dataset is tokenized according to full-word tokenization, where each orthographic word (i.e., sequence of letters separated by spaces) is treated as a single atomic token. In each dataset, the candidate morphological markers are treated as separate words, and so are likewise represented as single tokens. This choice makes the evaluation of both the language model and the theoretical morphological models easier, though it does place a strong limit on the applicability of these datasets to morphophonological analysis since the language models have no information about the phonological, or even orthographic, content of each token.

We study the behavior of decoder-only transformer models (Radford et al. 2018) with a model dimension $d_{\text{model}} = 512$, a feed-forward dimension of $d_{\text{ff}} = 4 \times d_{\text{model}} = 2048$, and $n_{\text{heads}} = 8$ attention heads, following established convention and the defaults specified in the PyTorch library (Paszke et al. 2019). Each decoder layer uses Gaussian Error Linear Units (Hendrycks & Gimpel 2016) as their non-linearity following their use in widespread transformer models like GPT-1 and BERT (Radford et al. 2018, Devlin et al. 2018). Each model has $n_{\text{layers}} = 6$ layers without dropout, resulting in a model with roughly 19 million trainable parameters.

Models are trained using the HuggingFace trainer on a single epoch of the training dataset (i.e., the model sees each sequence from the training dataset exactly once). Sequences are shuffled and stacked with a batch size of 32. Models are trained using the AdamW optimizer (Loshchilov & Hutter 2019) with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate with an initial value of $lr = 1 \times 10^{-4}$, decaying to 0 in a cosine schedule over the course of training. For each experimental condition we train 9 models with different random initializations (for a total of 720 independent models trained) and report the mean results per condition.

4 Results

4.1 Neural language models learn a sigmoid generalization

Figure 4.1 below shows the results of training transformer language models on the family of 80-noun direct-object datasets defined in the previous section. In general, the transformer language models' generalization probabilities for the regular marker $G_{T_{\text{LM}}}(m_{\text{reg}})$ increase as the productivity of the marker $r(m_{\text{reg}})$ increases. Between 10% and 50% productivity there is a considerable amount of variance in the generalization probabilities of individual models, indicating that models are quite sensitive to the random initialization of their parameters. Despite this, the mean-smoothed trend of generalization probability shows a fairly consistent, though not-quite monotonic increase in generalization probability with productivity; once $r(m_{\text{reg}})$ exceeds 60%, models generalize using m_{reg} nearly uniformly.

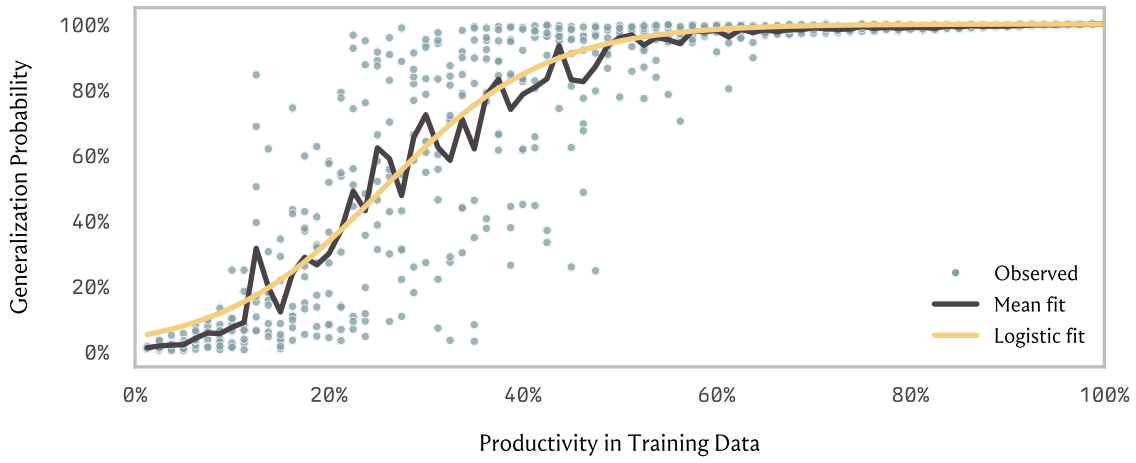


Figure 4.1: Scatterplot of measured mean generalization probabilities of transformer language models $G_{T_{\text{LM}}}(m_{\text{reg}})$ on the regular marker m_{reg} by the productivity $r(m_{\text{reg}})$ of their training datasets; lines show the mean and logistic fits to the observed data. As productivity increases, generalization probability increases as well, roughly following a sigmoid pattern.

The mean fit of the observed data closely follows an *S*-shaped sigmoid curve: generalization probability starts out quite low but increases rapidly until it reaches $\sim$ 50%, at which point the rate of increase slows until saturation near 100%. Logistic regression of the observed data shows that the mean generalization probability is well-approximated

by the sigmoid curve

$$G_{T_{\text{LM}}}(m_{\text{reg}}) \approx \frac{1}{1 + e^{-k(r(m_{\text{reg}}) - x_0)}}, \quad \text{where } k \approx 11.3 \text{ and } x_0 \approx 0.263.$$

Bootstrapped values for the logistic fit give 95% confidence intervals of $k \in [11.0, 11.7]$ and $x_0 \in [0.258, 0.270]$.

4.2 Formal models differ from neural language model behavior

The qualitative behavior of the transformer language models is, in general, quite different from that predicted by the various morphological productivity models used as numerical baselines, as show below in fig. 4.2. Unlike Yang's (2016) Tolerance Principle, the trained language models do not exhibit any sharp discontinuity in their generalization probabilities. While one could interpret the logistic relation between training productivity and generalization probability as being a soft approximation for a categorical jump between not generalizing and generalizing uniformly, the thresholds for the Tolerance Principle and the logistic regression of the neural language model data are not close: the Tolerance Principle predicts categorical generalization only when productivity exceeds 77.2% on our datasets, far outside the 95% confidence interval for the sigmoid fit's threshold value of $x_0 \approx 26.3\%$. Similarly, neural models exhibit near-categorical generalization when training productivity is only 50%.

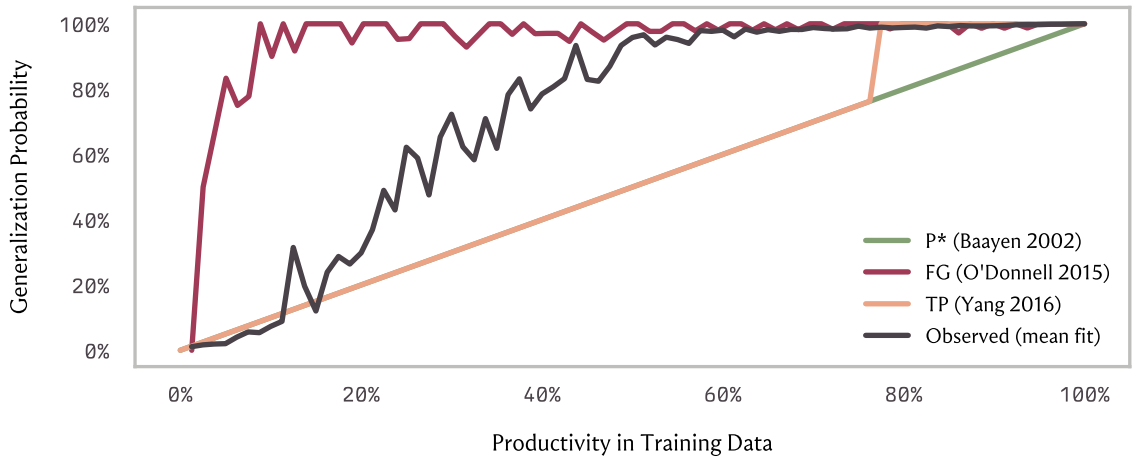


Figure 4.2: Qualitatively, the observed generalization behavior of language models does not match well with the predictions made by the P^*, the Productivity-Reuse Tradeoff, or Tolerance Principle models of morphological productivity.

Baayen's (2002) P^* appears to work well at predicting language model behavior when the productivity of the regular marker is low, at or below 10%; however, once $r(m_{\text{reg}}) > 10\%$, language models generalize m_{reg} with a far higher propensity that the P^* baseline would predict. Contrastingly, language models generalize to m_{reg} far less than O'Donnell's (2015) Productivity-Reuse Tradeoff model would expect, which predicts that a learner would use m_{reg} in novel contexts with near-uniformity once its productivity in the training corpora exceeded roughly 10%.

4.3 Language models generalize less strongly over time

The above data reports the mean generalization probability of models at the end of a single epoch of training. Over the course of training, neural models exhibited changes in their proclivity to generalize on datasets of a given productivity. In particular, we observe that models rapidly increase their mean generalization probability quite early on, followed by a modest decline from this peak over the remainder of training. The degree to which this overshooting occurs is closely predicted by the final generalization probability, and consequently also by training productivity: neural models, in general, don't overshoot when the productivity is near-zero; nor do models pass through a period of uniformly-generalizing without staying there over the duration of training. Rather, differences between mid-training and end-of-training generalization probabilities happen when models are trained on datasets with productivities ranging between 20% and 50%, as shown below in fig. 4.3. In these scenarios, though models may show close to a 10pt drop in mean generalization probability over the course of training, their qualitative remains unchanged: models still show a sigmoid-shaped curve over the course of training, though the slope parameter k changes as mean generalization probability decreases.

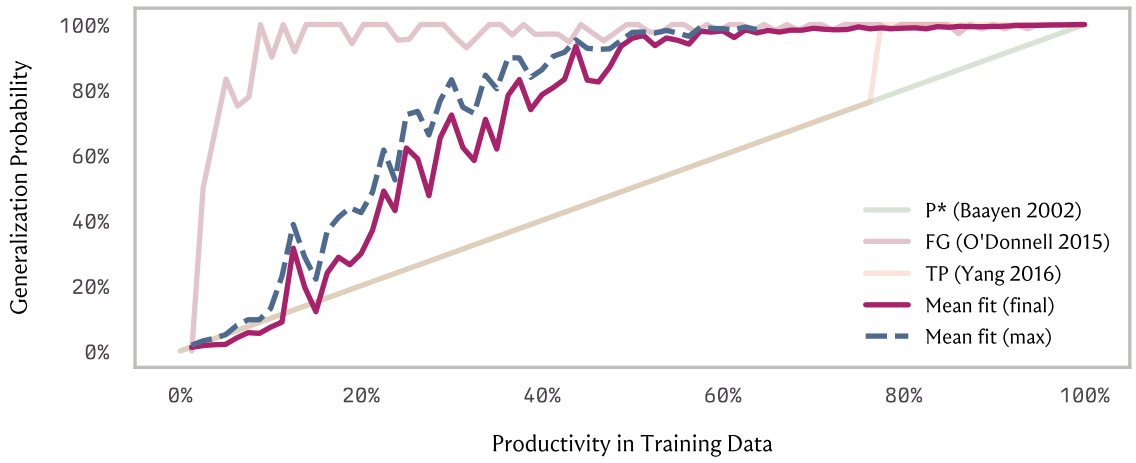


Figure 4.3: Comparing generalization probabilities at the points during training when models generalize with highest probability to that at the end of training shows that models generalize less strongly over the course of training.

5 Discussion

The experimental results presented above indicate that the mean generalization probability of the transformer language models follows an logistic or 'sigmoid' curve (the 1-dimensional version of the softmax function) with respect to the underlying productivity of a morpheme in training: for every unit increase in a morpheme's usage in the training distribution, it's generalization probability increases like

$$G_{T_{\text{LM}}}(m_{\text{reg}}) \sim \frac{1}{1 + e^{-r(m_{\text{reg}})}} = \text{softmax}(r(m_{\text{reg}})).$$

Here, I argue that these empirical results are both *mathematically interpretable* and *linguistically interesting*. They are interpretable in the sense that we can provide at least a partial explanation for what causes the transformers to behave in this manner; they are interesting in the sense that this behavior, though qualitatively unlike the predictions made by the three models of morphology surveyed above, is nevertheless well-attested in the predictions of other formal models of linguistic phenomena and that this correspondence follows from the overlap between the methods of neural optimization employed during language model training and the assumptions made in deriving maximum-entropy models of linguistic phenomena.

5.1 Why do language models learn a sigmoid generalization?

Since the language models' generalization probabilities are posterior conditional probability distributions, they are related by Bayes' theorem to the respective likelihood, prior, and marginal distributions in the following way:

$$T_{\text{LM}}(t_i \mid w) = P(t_i \mid w) = \frac{P(w \mid t_i)P(t_i)}{P(w)}.$$

The marginal term $P(w)$ here is merely a normalizing factor which is constant with respect to the productivity of t_i. Similarly, by construction the various morphological markers t_i are not predictive of their conditioning environments because there are no lexical- or morphemic features which condition their use. As such, the likelihood term $P(w \mid t_i)$ is likewise constant in the frequency of t_i. The only term in the Bayes' decomposition which is sensitive to the underlying frequency of t_i is the prior term $P(t_i)$, which encodes a model's believes on the unconditional probability of t_i being used. As such we can interpret the models' generalization probabilities in terms of their learned priors:

$$G_{T_{\text{LM}}}(t_i) \propto T_{\text{LM}}(t_i \mid w) \propto P(t_i).$$

Hence, the observed behavior of the neural models shows that they learn a sigmoidal prior for the morphological markers as a function of those marker's productivity:

$$P(m_{\text{reg}}) \sim \text{softmax}(r(m_{\text{reg}})).$$

One interpretation for the sigmoidal relationship between a model's prior for m_{reg} and the productivity $r(m_{\text{reg}})$ is that the model exhibits *calibration error*: counterfactually, any statistical model's estimated class probabilities should (in a normative sense) be consistent with the underlying data the model is trained on. A model minimizes the gap between observed class frequencies predicted probabilities on a set of validation data which has been drawn *i.i.d.* from the same distribution as the training data by simply 'passing through' the class frequency as its output probability. Such a model exhibits a linear relationship between productivity and generalization probability. Under the assumptions of the datasets considered here, where classes are not predictive of their conditioning contexts, this would result from a model who's learned prior is *linear* in class frequency. Yet empirically, we observe models whose learned priors are *sigmoidal* in class frequency.

Under the interpretation that the observed behavior is indicative of miscalibration, it is worth pondering why this behavior should emerge. One plausible explanation is that such patterns arise when models overfit to their training data. For arbitrary statistical classifiers, calibration can be ensured by stopping training once loss on an *i.i.d.* validation set begins to diverge from loss on the training distribution. There is not guarantee that this divergence will happen at the same time on two different datasets if the underlying distributions are different. Indeed, we should expect that models trained on datasets with higher entropy should experience validation loss divergence earlier than those trained on datasets with lower entropy, all else being equal. Figure 5.1 below demonstrates this effect empirically in the toy setting of binary classification on randomly-generated biased data. With early stopping, models train longest on low-entropy datasets (when class frequency is lowest or highest) and shortest on high entropy datasets (when classes are equally balanced). In this setting, models become well-calibrated and show a linear relationship between class frequency and generalization probability. When trained without early stopping, each model sees the same amount of data regardless of entropy or validation loss divergence. Here, models become miscalibrated, showing a sigmoid relationship between class frequency and generalization probability.

Yet unlike the toy classifier example, our transformer language models *do not* exhibit validation loss divergence on any datasets; rather, all models show validation loss decreasing monotonically over the course of training. This is unsurprising given that our models are trained for only a single epoch, and so see each unique sequence only once, making overfitting implausible. The reconcilability between these two points is most likely attributable to the differences in training objective between a binary classifier and our language models. While both kinds of models are trained to minimize cross-entropy loss, the binary classifiers detailed above receive loss signal on only a single predictive task: is the input of class *A* or *B*; by contrast, language models must learn to be good *overall* models of the sequences they encounter without privileging one particular aspect of sequence well-formedness over any other. As discussed in section 2, this impartiality to any single linguistic task makes language models a better fit for modeling acquisition than other kinds of neural network. It also permits models to exhibit what look like calibration errors without experiencing loss divergence: since the contribution of the morpheme generalization task examined here to the model's overall loss is small relative to the combined loss from every other aspect of producing well-formed sequences, a model's overall validation loss can continually decrease while its calibration on this particular task can diverge.

As purely statistical models, the observed non-linear relationship between underlying productivity and general-

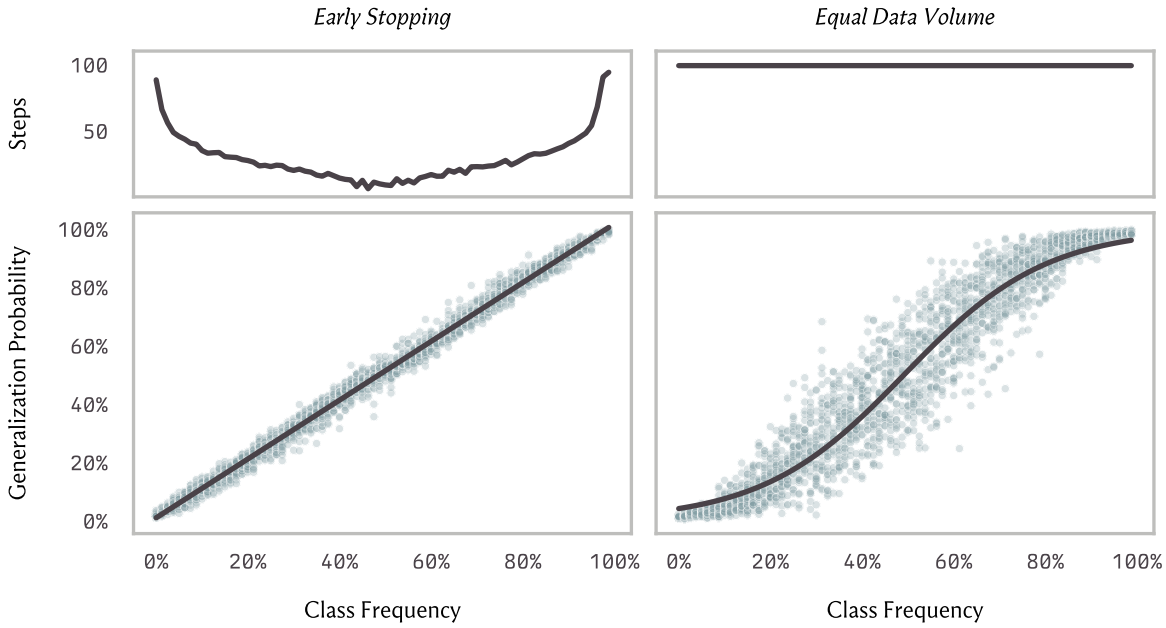


Figure 5.1: Multi-layer dense binary classifiers can be induced to exhibit different relationships between generalization probability and class frequency depending on training conditions. **(left)** When trained with early stopping, models are well-calibrated and exhibit a linear relationship between generalization probability and class frequency. **(right)** When trained on a fixed amount of data (i.e., without early stopping) models exhibit a sigmoid relationship between generalization probability and class frequency.

ization probability is reasonably analyzed as calibration error. But as models of linguistic acquisition, it is far less clear that the observed behavior is truly 'erroneous.' The argument from ecological plausibility extends beyond the training objective of a model: it also informs the training dynamics. Human language learners do not stop receiving linguistic over the course of their lives, nor do they stop learning overall from their input conditional on learning well any particular linguistic phenomenon. Given this reality, attempting to mitigate calibration error in a model of acquisition by conditioning training on something akin to validation loss divergence would not be desirable. As such, the lack of linearity observed is not an error merely because it is indicative of miscalibration; it is only erroneous if it is poorly predictive of human behavior.

5.2 Language models as MEHG learners

In seeking to understand why language models generalize morpheme use as a sigmoid function of productivity, it is worth noting that this qualitative relationship is well attested in other linguistic theories. In particular, the same *S*-shaped logistic curves seen above also appear prolifically in the context of Maximum Entropy Harmonic Grammar (MEHG; Goldwater & Johnson 2003), an extension of Prince & Smolensky's (2004) Optimality Theory which grades potential forms probabalistically as a function of their underlying harmony with respect to posited constraints. The appearance of sigmoid relationships in MEHG models is by construction, where the conditional

probability of some output form is defined as

$$P(y \mid x) = \frac{\exp\left(\sum_j w_j f_j(y,x)\right)}{\sum_y \exp\left(\sum_j w_j f_j(y,x)\right)}, \quad \text{where } w_i \text{ weight discrete featural functions } f_j(y,x).$$

Among competing outputs y, we may define the inner sum over weighted featural scores to be the *harmony* $h(y,x) = \sum_j w_j f_j(y,x)$ of y in the context of x. This recovers the expected sigmoid relationship between output probability and harmony:

(13) $P(y \mid x) = \text{softmax}(h(y,x))$.

Importantly, the MEHG models of Goldwater & Johnson (2003) place one additional stipulation on the harmony term $h(y,x)$ which is not present for maximum entropy models in general: to apply the principle of maximum entropy to an Optimality Theory framework, Goldwater & Johnson constrain the featural functions $f_j(y,x)$ to be *linear* functions of the underlying scalar scores for the pair (y,x); in particular, they take $f_j(y,x)$ to be the number of violations (y,x) incurs for a constraint C_j. This linear relationship between harmony and gradable input features is necessary to guarantee that MEHG models exhibit a sigmoid-relationship between input features and output probability; counterfactually, if harmony is not at least affine in a relevant input feature then there is no reason to expect an *S*-shaped curve to arise, as is shown below in fig. 5.2.

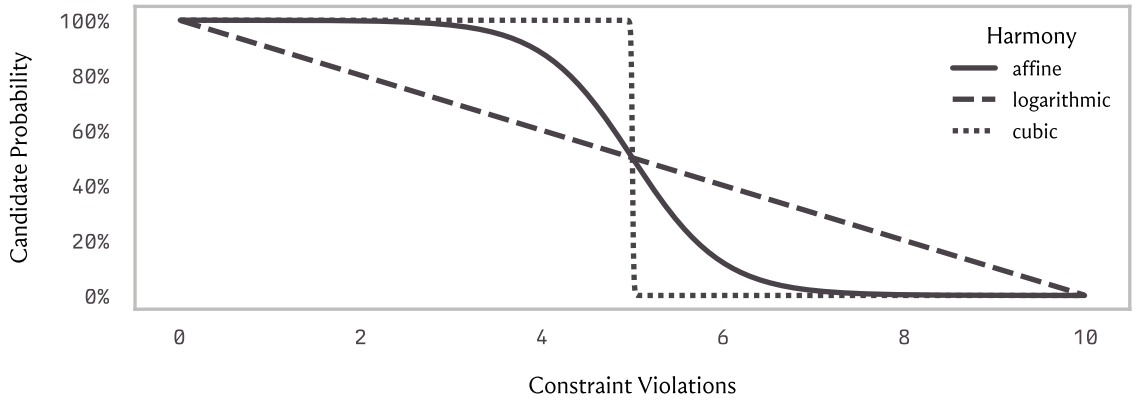


Figure 5.2: When harmony is an affine function of underlying features, MEHG models show an *S*-shaped sigmoid curve; when harmony is logarithmic, models show a linear relationship; when harmony is cubic, models approach a centered indicator function.

Although originally formulated for generative phonology, Hayes (2021) notes that the quantitative signature of maximum entropy models—a sigmoid relationship between observed use and some predictive variable—appears widely in domains not typically modeled using constraint-based frameworks, including phonetic (consonant voicing conditioned on vowel length; Kluender, Diehl & Wright 1988), syntactic (dative and genitive construction frequency; Szmrecsanyi et al. 2017), and historical linguistic phenomena (frequency of use of an innovative form over time; Kroch 1989, Zimmermann 2017). Hayes (2021) notes in a concluding remark that the prevalence of these sigmoid relations in disparate linguistic phenomenon is notable for the fact that competing generative theories do not yield the same quantitative signature; as such, the abundance of sigmoid-shaped phenomena makes MEHG compelling

framework in which to formulate theories of linguistic processes. While Hayes (2021) does not cite examples from morphology in his survey of maximum entropy, it is not difficult to formulate such a theory to model morphological productivity:

(14) **An MEHG model of morphological productivity**
Given a competing set of licit morphological markers $\{t_i\}$ for a novel word w, a learner will use t_i with probability $P(t_i \mid w) = \text{softmax}(h(t_i, w))$, where the harmony $h(t_i, w)$ is an affine function of t_i's relative productivity in the learner's previous linguistic input.

While such a formal model of morphological productivity is conceivable, it is an open question as to whether or not it would be well supported by human data. I am not aware of any existing proposed theories of morphological productivity which make use of a maximum entropy harmonic grammar model, though Müller (2019) does propose the use of an augmented form of (regular) harmonic grammar to describe some morphological and syntactic phenomena, noting that it can in principle be combined with the maximum entropy framework to produce probabalistic results.

What is presently demonstrable is that the maximum entropy harmonic grammar models proposed by Goldwater & Johnson (2003) *are* a good fit for the observed data of neural language models. In one sense, this is unsurprising: maximum entropy models in general predate their use as models of grammar, and are motivated by principles from information theory without regard for specific aspects of the language faculty (Jaynes 1958, Hayes 2021). Maximum entropy models found particular purchase in the connectionist models of cognition and language (see, *inter alia*, Feldman & Ballard 1982, Jordan 1986, Elman 1990). Nearly all neural models used today are, by construction, maximum entropy models by virtue of their design as featural classifiers gated by a softmax function (recall definition 2.2.3) and training objective to minimize cross-entropy loss.[7] Indeed, we can even explicitly interpret the inputs to a neural model's softmax function in terms of its learned likelihoods and priors. Again by Bayes' theorem we know that a model's posterior distribution is analyzable in terms of its likelihood, prior, and marginal distributions; expanding slightly the marginal term, we see that

$$T_{\text{LM}}(t_i \mid w) = \frac{P(w \mid t_i)P(t_i)}{P(w)} = \frac{P(w \mid t_i)P(t_i)}{\sum_j P(w \mid t_j)P(t_j)}.$$

The role of the softmax classifier in the model is made explicit through a clever application of exponentiation. Letting $h(t_i, w) \triangleq \ln P(w \mid t_i) + \ln P(t_i)$, we see that

$$T_{\text{LM}}(t_i \mid w) = \frac{\exp h(t_i, w)}{\sum_j \exp h(t_j, w)} = \text{softmax}(h(t_i, w)). \tag{16}$$

As the intentional choice of variable naming in (16) may suggest, this derivation coincides exactly with the form of maximum entropy harmonic grammar models presented above in (13); consequently, we can interpret a neural model's unnormalized logits as a form of harmony, decomposable by Bayes' theorem into a sum of log-likelihood and log-prior terms.

While the above derivation is merely a consequence of the design of neural models, the experimental results obtained in this study reveal something unexpected and non-obvious. As demonstrated previously in fig. 5.1, the

7. Somewhat confusingly, the principle of training models to *minimize cross-entropy* (Kullback 1959) is a generalization of Jaynes's (1958) principle of *maximum entropy* to conditions when the prior distribution is known; see Shore & Johnson (1980) for a more thorough treatment of the connection.

mere fact that neural models have the structure of (16) is not sufficient to produce sigmoid-shaped curves; indeed, it is the normal condition of well-calibrated neural models to exhibit a strongly linear relationship between input features and output probability. The sigmoid relationship observed in the miscalibrated binary classifiers in fig. 5.1 or in the neural language models examined here can only arise when the harmony term $h(t_i, w)$ of (16) is itself an affine function of the relevant input feature. This is exactly the stipulation that Goldwater & Johnson (2003) propose for their MEHG models. Consequently, the empirical behavior of the neural language models shows that they are not just maximum entropy models but specifically MEHG models of the form described in (14). It is therefore reasonable to speculate that although none of the formal models of morphological productivity examined here are good predictors of neural language model behavior, an MEHG model along the lines of (14) *would* be a good fit for neural language models while *also* being a reasonable hypothesis for a model of human behavior by analogy to the success of MEHG models proposed for adjacent linguistic phenomena.

5.3 Limitations & future work

5.3.1 Comparing language model behavior to human data

The present study compares the behavior of neural language models to predictions made by formal theories of morphological productivity. We largely answer this question in the negative—among the three formal models surveyed here (Baayen's (2002) P^*, Yang's (2016) Tolerance Principle, and O'Donnell's (2015) Fragment Grammars), none are particularly good quantitative or even qualitative fits for the behavior of neural models. Interestingly, though all formal models disagree with the behavior of the neural models, they also disagree with each other: while there is a general agreement that when productivity is near-zero a human learner will never generalize and that the generalization probability is roughly-monotonically increasing in productivity, each of the three formal models quickly diverge from one another for significant portions of the input space. This lack of consensus raises an interesting point: how do human language learners generalize when exposed to the kinds of datasets constructed here? In light of the discussion above, this can be refined into to specific questions: first, empirically, are neural language models good models of human behavior on this morphological generalization task? And second, more formally, is the behavior of human language learners well-modelled by an MEHG model of morphological productivity (which, contingently, is also good model for the neural models explored here)?

Answering these two questions requires additional investigation since the present study did not conduct comparable experiments with human subjects. The most straightforward experimental design for this would be to extend the artificial language learning (ALL) experiments detailed in Schuler, Yang & Newport (2016) and Culbertson & Schuler (2019), wherein human participants are exposed to datasets of fragmentary artificial languages instantiating different morphological patterns and then tested on a generalization set. In particular, conducting ALL experiments on the parametrically-varied datasets created here would allow for a direct comparison between human and language model behavior. While Schuler, Yang & Newport (2016), in seeking to experimentally test the Tolerance Principle, do technically measure the impact of morpheme productivity on generalization behavior, their setup is constrained in ways which limit its use in answering these questions.[8]

8. First, they consider datasets with only 9 distinct inflectional targets (compared to the 80 used here) which greatly coarsens the space of featural variation, especially compared to the kinds of morphological processes which theories like the Tolerance Principle are designed to model (e.g., there are far more than 9 nouns which serve as targets for plural morphology). Second, among this already constrained space, they sample only two points to study: one dataset which is below the Tolerance Principle's predicted threshold (3 regular nouns and 6 exceptional ones), and one above it (5 regular nouns and 4 exceptional ones). These points are not evenly spaced around the Tolerance Principle's threshold

5.3.2 Refining tokenization to include featural representation

This study examined how language models generalize a morphological distribution when treating morphemes as atomic combinatorial pieces: by construction, the word-level tokenization employed here treats each unique space-separate sequence of symbols as a distinct token in a language model's vocabulary, as shown in (16) below. This decision greatly simplifies the experimental procedure in two ways: first, it shortens the sequence length of each example, reducing the computational overhead for the model; second, it controls for any other representational features which could influence how a learner generalizes. While the former is unconditionally helpful, the latter may not be. Indeed, though the theoretical models of morphological productivity surveyed here are agnostic to the representational features of the morphemes in question, such as their phonology or etymology, human learners are not. Other theoretical models of morphological learning, such as Albright & Hayes's (2002) Minimum Generalization Learner (MGL), explicitly incorporate such information. Refining the tokenization scheme to include either orthographic or phonological information would increase the ecological plausibility of the training and evaluation setup and permit comparison between the behavior of neural language models and the predictions of a broader array of formal models of morphological learning.

(16)

word-level	*orthographic*	*phonological*
·the·cats·	·t·h·e· ·c·a·t·s·	·ð·ə· ·k·æ·t·s·

There are two broad categories of experiment that should be done with a refined tokenization scheme. First, replicating the current experimental procedure with tokens representing orthographic or phonological features would show the degree to which morpheme-level features influence generalization behavior beyond frequency effects. Second, separately from the influence of frequency effects, training language models with morpheme-level features would demonstrate whether or not neural models can serve as plausible models for word-internal morphological generalization by comparing their behavior on *wug*-test tasks like the SIGMORPHON-UniMorph 2021 task on word reinflection (Pimentel et al. 2021), as Wilson & Li (2021) do for featurally-aware formal models like the MGL.

5.3.3 Investigating more morphological phenomena

We investigate here morphological productivity in the context of direct object marking, but in principle the datasets we construct are agnostic to the particular morphological phenomena in question so long as they can be represented in a context-free grammar. Further investigation should explore whether similar behavior arises when models are trained on datasets exemplifying other morphological phenomena, such as plural marking. Similarly, we should investigate whether models exhibit similar behavior when training datasets contain multiple different morphological processes, as opposed to the highly restricted setting here where only one is present.

value, and also happen to overlap with the majority boundary which is a contingent property of there being so few nouns.

6 Conclusion

During language acquisition, humans learn to generalize morphological patterns to novel forms in systematic ways. Many formal models of morphological productivity argue that human learners generalize in part on distributional properties of their linguistic input; specifically, the frequency of a morpheme's occurrence in one's linguistic input is predictive of the likelihood of that morphemes use in a novel context.

We examine whether the generalization behavior of neural language models is predicted by any of three formal models of morphological productivity: Baayen's (2002) P^*, O'Donnell's (2015) Productivity-Reuse Tradeoff, and Yang's (2016) Tolerance Principle. To compare neural language model behavior against the predictions made by these formal models, we construct formal grammars of a fragmentary version of English which uses overt direct-object marking. We then generate datasets which parametrically vary the productivity of a given object marker by sampling from these grammars. Given the distributional properties of morpheme use within each dataset, we construct numerical baselines based on how the three formal models of productivity predict a human learner would generalize the given marker's use based on its observed productivity. We then train transformer language models on these datasets and measure their propensity to generalize the given marker to novel contexts and compare this behavior to the predictions made by the formal models.

Our main conclusion is negative: none of the three formal models is well-predictive of neural language model behavior: while the Productivity-Reuse Tradeoff and the Tolerance Principle predict learners will form categorical generalizations for morpheme use (though they disagree at what point this occurs) and Baayen's P^* predicts a morpheme's use linearly increases with its input frequency, the transformer language models trained here show a sigmoid-relation between generalization probability and input productivity. To better understand why the neural models learn this kind of generalization, we decompose model behavior using a basic application of Bayes' theorem to conclude that the sigmoid-relation between generalization probability and productivity arises as a contingent property of a neural language model's training dynamics; specifically, we argue that the observed behavior arises due to a particular form of miscalibration, where the models' learned log-prior distribution over next-tokens is affine (rather than the expected logarithmic) in relative frequency.

We connect this empirical result back to linguistic theory by demonstrating that this affine log-prior relation is characteristic of a formal linguistic framework known as Maximum Entropy Harmony Grammar. In particular, we note that since the *harmony* measure used to score candidates in MEHG models also obeys an affine relation to gradable input features, neural language models can be analyzed not just as generalized maximum entropy models but more specifically as highly-parameterized MEHG models. We use these insights to predict that though the three formal models of morphological productivity selected as baselines serve as good predictors of language model behavior, a formal MEHG model of productivity would do so quite well while being plausible from first principles as a hypothesis for human behavior as well. We conclude by discussing the limitations of the present study (in

particular, that additional studies with human subjects are needed to test any correspondence between language model and human behavior) and present avenues for future work to remedy these shortcomings.

Bibliography

Albright, Adam & Bruce Hayes. 2002. Modeling english past tense intuitions with minimal generalization. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning -*, 58–69. Morristown, NJ, USA: Association for Computational Linguistics. https://doi.org/10.3115/1118647.1118654.

Anthropic. 2023. *Claude 2*. https://web.archive.org/web/20230811143536/https://www.anthropic.com/index/claude-2.

Autebert, Jean-Michel, Jean Berstel & Luc Boasson. 1997. Context-free languages and pushdown automata. In *Handbook of Formal Languages*, 111–174. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-59136-5_3.

Baayen, R Harald. 1989. *A Corpus-Based approach to morphological productivity: Statistical analysis and psycholinguistic interpretation*. Amsterdam: Vrije Univertsiteit dissertation.

Baayen, R Harald. 1992. Quantitative aspects of morphological productivity. *Yearbook of Morphology*. 109–149. https://doi.org/10.1007/978-94-011-2516-1_8.

Baayen, R Harald. 2002. *Word Frequency Distributions* (Text, Speech and Language Technology). New York, NY: Springer. 335 pp. https://doi.org/10.1007/978-94-010-0844-0.

Baayen, R Harald. 2009. Corpus linguistics in morphology: Morphological productivity. *Corpus Linguistics*. 899–919. https://doi.org/10.1515/9783110213881.2.899.

Bauer, Laurie. 2006. *Cambridge studies in linguistics: Morphological productivity series number 95*. Cambridge, England: Cambridge University Press. 260 pp. https://doi.org/10.1017/cbo9780511486210.

Berko, Jean. 1958. The child's learning of English morphology. *WORD* 14. 150–177. https://doi.org/10.1080/00437956.1958.11659661.

Biderman, Stella, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika & Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *arXiv [cs.CL]*. eprint: 2304.01373 (cs.CL).

Corkery, Maria, Yevgen Matusevych & Sharon Goldwater. 2019. Are we there yet? Encoder-decoder neural networks as cognitive models of English past tense inflection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3868–3877. Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/p19-1376.

Cotterell, Ryan, Anej Svete, Clara Meister, Tianyu Liu & Li Du. 2023. Formal aspects of language modeling. *arXiv [cs.CL]*. eprint: 2311.04329 (cs.CL).

Culbertson, Jennifer & Kathryn Schuler. 2019. Artificial language learning in children. *Annual review of linguistics* 5. 353–373. https://doi.org/10.1146/annurev-linguistics-011718-012329.

Dankers, Verna, Anna Langedijk, Kate McCurdy, Adina Williams & Dieuwke Hupkes. 2021. Generalising to German plural noun classes, from the perspective of a recurrent neural network. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, 94–108. Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.conll-1.8.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv [cs.CL]*. eprint: 1810.04805 (cs.CL).

Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive science* 14. 179–211. https://doi.org/10.1016/0364-0213(90)90002-E.

Feldman, J A & D H Ballard. 1982. Connectionist models and their properties. *Cognitive science* 6. 205–254. https://doi.org/10.1207/s15516709cog0603_1.

Gemini Team et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv [cs.CL]*. eprint: 2312.11805 (cs.CL).

Goldwater, S & Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the workshop on variation within Optimality Theory*, 111–120. https://web.science.mq.edu.au/~mjohnson/papers/GoldwaterJohnson03.pdf.

Groeneveld, Dirk, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A Smith & Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. *arXiv [cs.CL]*. eprint: 2402.00838 (cs.CL).

Hayes, Bruce. 2021. Deriving the Wug-shaped curve: A criterion for assessing formal theories of linguistic variation. https://linguistics.ucla.edu/people/hayes/papers/HayesWugShapedCurve2021ShortVersion.pdf.

Hendrycks, Dan & Kevin Gimpel. 2016. Gaussian Error Linear Units (GELUs). *arXiv [cs.LG]*. eprint: 1606.08415 (cs.LG).

Hudson Kam, Carla L & Elissa L Newport. 2005. Regularizing unpredictable variation: The roles of adult and child learners in language formation and change. *Language learning and development: the official journal of the Society for Language Development* 1. 151–195. https://doi.org/10.1080/15475441.2005.9684215.

Hudson Kam, Carla L & Elissa L Newport. 2009. Getting it right by getting it wrong: when learners change languages. *Cognitive psychology* 59. 30–66. https://doi.org/10.1016/j.cogpsych.2009.01.001.

Hupkes, Dieuwke, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell & Zhijing Jin. 2022. State-of-the-art generalisation research in NLP: A taxonomy and review. *arXiv [cs.CL]*. eprint: 2210.03050 (cs.CL).

Jaynes, Edwin Thompson. 1958. Information Theory and Statistical Mechanics. *Physical review* 106. 620–630. https://doi.org/10.1103/PhysRev.106.620.

Johnson, Mark, Thomas L Griffiths & Sharon Goldwater. 2007a. Adaptor Grammars: A Framework for Specifying Compositional Nonparametric Bayesian Models. In *Advances in Neural Information Processing Systems 19*, 641–648. MIT Press. https://www.research.ed.ac.uk/files/12435989/NIPS2006_0064.pdf.

Johnson, Mark, Thomas L Griffiths & Sharon Goldwater. 2007b. Bayesian Inference for PCFGs via Markov Chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 139–146. https://aclanthology.org/N07-1018.pdf.

Jordan, Michael I. 1986. *Serial order: a parallel distributed processing approach*. Research rep. 8604. Institute for Cognitive Science, University of Californa, San Diego. 40 pp. https://cseweb.ucsd.edu/~gary/PAPER-SUGGESTIONS/Jordan-TR-8604-OCRed.pdf.

Jurafsky, Dan & James Martin. 2008. *Speech and Language Processing*. 2nd edn. Upper Saddle River, NJ: Pearson. 1032 pp.

Kirov, Christo & 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. https://doi.org/10.1162/tacl_a_00247.

Kluender, Keith R, Randy L Diehl & Beverly A Wright. 1988. Vowel-length differences before voiced and voiceless consonants: an auditory explanation. *Journal of phonetics* 16. 153–169. https://doi.org/10.1016/s0095-4470(19)30480-2.

Kroch, Anthony S. 1989. Reflexes of grammar in patterns of language change. *Language variation and change* 1. 199–244. https://doi.org/10.1017/s0954394500000168.

Kullback, Solomon. 1959. *Information theory and statistics*. 399 pp.

Liu, Ling & Mans Hulden. 2022. Can a transformer pass the wug test? Tuning copying bias in neural morphological inflection models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 739–749. Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-short.84.

Loshchilov, Ilya & Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7.

Malouf, Robert. 2017. Abstractive morphological learning with a recurrent neural network. *Morphology* 27. 431–458. https://doi.org/10.1007/s11525-017-9307-x.

Marantz, Alec. 2001. Words. https://babel.ucsc.edu/~hank/mrg.readings/Marantz_words.pdf.

Marcus, G F. 1995. Children's overregularization of English plurals: a quantitative analysis. *Journal of child language* 22. 447–459. https://doi.org/10.1017/s0305000900009879.

Marcus, G F, U Brinkmann, H Clahsen, R Wiese & S Pinker. 1995. German inflection: the exception that proves the rule. *Cognitive psychology* 29. 189–256. https://doi.org/10.1006/cogp.1995.1015.

McCurdy, Kate, Sharon Goldwater & Adam Lopez. 2020. Inflecting when there's no majority: Limitations of encoder-decoder neural networks as cognitive models for German plurals. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1745–1756. Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.159.

Meta AI. 2024. *Introducing Meta Llama 3: The most capable openly available LLM to date*. https://ai.meta.com/blog/meta-llama-3/.

Müller, Gereon. 2019. *Gradient Harmonic Grammar*. https://home.uni-leipzig.de/muellerg/c25.pdf.

O'Donnell, T, Noah D Goodman & J Tenenbaum. 2009. Fragment grammars: Exploring computation and reuse in language. https://sites.socsci.uci.edu/~lpearl/colareadinggroup/readings/ODonnellEtAl2009_FragmentGrammars.pdf.

O'Donnell, Timothy J. 2015. *Productivity and reuse in language* (The MIT Press). London, England: MIT Press. 350 pp. https://doi.org/10.7551/mitpress/9780262028844.001.0001.

OpenAI. 2023. GPT-4 Technical Report. *arXiv [cs.CL]*. eprint: 2303.08774 (cs.CL).

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, N Gimelshein, L Antiga, Alban Desmaison, Andreas Köpf, E Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai & Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. *Neural Information Processing Systems* abs/1912.01703. 8026–8037. https://doi.org/10.5555/3454287.3455008.

Pimentel, Tiago, Maria Ryskina, Sabrina J Mielke, Shijie Wu, Eleanor Chodroff, Brian Leonard, Garrett Nicolai, Yustinus Ghanggo Ate, Salam Khalifa, Nizar Habash, Charbel El-Khaissi, Omer Goldman, Michael Gasser, William Lane, Matt Coler, Arturo Oncevay, Jaime Rafael Montoya Samame, Gema Celeste Silva Villegas, Adam Ek, Jean-Philippe Bernardy, Andrey Shcherbakov, Aziyana Bayyr-ool, Karina Sheifer, Sofya Ganieva, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Andrew Krizhanovsky, Natalia Krizhanovsky, Clara Vania, Sardana Ivanova, Aelita Salchak, Christopher Straughn, Zoey Liu, Jonathan North Washington, Duygu Ataman, Witold Kieraś, Marcin Woliński, Totok Suhardijanto, Niklas Stoehr, Zahroh Nuriah, Shyam Ratan, Francis M Tyers, Edoardo M Ponti, Grant Aiton, Richard J Hatcher, Emily Prud'hommeaux, Ritesh Kumar, Mans Hulden, Botond Barta, Dorina Lakatos, Gábor Szolnok, Judit Ács, Mohit Raj, David Yarowsky, Ryan Cotterell, Ben Ambridge & Ekaterina Vylomova. 2021. SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 229–259. Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.sigmorphon-1.25.

Prince, Alan & Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. 1st edition. Malden, MA: Wiley-Blackwell. 304 pp.

Radford, Alec, Karthik Narasimhan, Tim Salimans & Ilya Sutskever. 2018. *Improving language understanding by generative pre-training*. Tech. rep. OpenAI. 12 pp. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Schuler, Kathryn, Charles Yang & Elissa Newport. 2016. Testing the Tolerance Principle: Children form productive rules when it is more computationally efficient to do so. In Anna Papagragou, Daniel Grodner, Daniel Mirman & John Trueswell (eds.), *Proceedings of the 38th annual meeting of the cognitive science society*, 2321–2326. Austin, TX: Cognitive Science Society. https://www.ling.upenn.edu/~ycharles/papers/syn2016.pdf.

Shore, J & R Johnson. 1980. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE transactions on information theory* 26. 26–37. https://doi.org/10.1109/tit.1980.1056144.

Szmrecsanyi, Benedikt, Jason Grafmiller, Joan Bresnan, Anette Rosenbach, Sali Tagliamonte & Simon Todd. 2017. Spoken syntax in a comparative perspective: The dative and genitive alternation in varieties of English. *Glossa a journal of general linguistics* 2. 86. https://doi.org/10.5334/gjgl.310.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (NIPS'17), 6000–6010. Red Hook, NY, USA: Curran Associates Inc.

Weissweiler, Leonie, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer & David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 6508–6524. Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.401.

Wilson, Colin & Jane S Y Li. 2021. Were we there already? Applying minimal generalization to the SIGMORPHON-UniMorph shared task on cognitively plausible morphological inflection. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 283–291. Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.sigmorphon-1.29.

Yang, Charles. 2016. *The Price of Linguistic Productivity: How Children Learn to Break the Rules of Language*. MIT Press. 280 pp.

Zimmermann, Richard. 2017. *Formal and quantitative approaches to the study of syntactic change: three case studies from the history of English*. https://doi.org/10.13097/ARCHIVE-OUVERTE/UNIGE:96500.

Zipf, George Kingsley. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Cambridge, MA: Addison-Wesley Press.

A Example Grammars

Training Grammar (1/80)

```
S -> NP VP
NP -> DET SUBJ
SUBJ -> "tangerine" | "violin" | "vase" | "whistle" | "sunset" | "yarn" | "ivory"
    | "quasar" | "notebook" | "elephant" | "gazelle" | "harp" | "mosquito" | "xylum"
    | "feather" | "cactus" | "rosemary" | "parrot" | "gelato" | "saxophone" | "coffee"
    | "candle" | "tree" | "nougat" | "rattlesnake" | "guitar" | "jacket" | "zenith"
    | "quiver" | "envelope" | "flower" | "accordion" | "upholstery" | "vortex"
    | "xenon" | "apple" | "banana" | "quilt" | "kettle" | "apricot" | "shovel"
    | "dog" | "penguin" | "sandcastle" | "xylophone" | "ocean" | "muffin"
    | "kangaroo" | "jigsaw" | "puzzle" | "honey" | "wallet" | "lemonade" | "zebra"
    | "lawyer" | "fountain" | "volcano" | "nutmeg" | "mountain" | "rocket"
    | "quill" | "kayak" | "jungle" | "lamp" | "whale" | "puffin" | "airplane"
    | "telescope" | "zeppelin" | "mango" | "kiwi" | "cat" | "xerox" | "umbrella"
    | "lighthouse" | "bellflower" | "drum" | "dragonfly" | "trampoline" | "fireplace"
    | "unicorn" | "giraffe" | "desk" | "eagle" | "yacht"
VP -> VERB DET OBJ | ADV VERB DET OBJ
ADV -> "from time to time"
VERB -> "sees" | "hears" | "resembles" | "likes" | "dislikes"
DET -> "the" | "a"
OBJ -> REG_OBJ | IRR_OBJ
REG_OBJ -> "yarn" "DO"
IRR_OBJ -> "ivory" "cyp" | "quasar" "ikm" | "notebook" "wvm" | "elephant" "jjb"
    | "gazelle" "ujy" | "harp" "jxu" | "mosquito" "aya" | "xylum" "nyp"
    | "feather" "ctq" | "cactus" "lvq" | "rosemary" "sjp" | "parrot" "mek"
    | "gelato" "sbf" | "saxophone" "gcz" | "coffee" "xxv" | "candle" "keu"
    | "tree" "sne" | "nougat" "qwb" | "rattlesnake" "lwl" | "guitar" "gfd"
    | "jacket" "vkz" | "zenith" "kxe" | "quiver" "jjq" | "envelope" "cke"
    | "flower" "ogx" | "accordion" "lim" | "upholstery" "ndy" | "vortex" "vgv"
    | "xenon" "zha" | "apple" "rqd" | "banana" "luj" | "quilt" "usx"
    | "kettle" "fvt" | "apricot" "acr" | "shovel" "kbr" | "dog" "byp"
```

```
    | "penguin" "man" | "sandcastle" "smo" | "xylophone" "ojc" | "ocean" "lac"
    | "muffin" "gej" | "kangaroo" "zvc" | "jigsaw" "ksr" | "puzzle" "rvl"
    | "honey" "xju" | "wallet" "tdu" | "lemonade" "zji" | "zebra" "csw"
    | "lawyer" "sri" | "fountain" "xvw" | "volcano" "xnx" | "nutmeg" "ctn"
    | "mountain" "nnu" | "rocket" "dbh" | "quill" "yup" | "kayak" "yaf"
    | "jungle" "bll" | "lamp" "nxj" | "whale" "jxy" | "puffin" "uct"
    | "airplane" "ndo" | "telescope" "cex" | "zeppelin" "kod" | "mango" "yab"
    | "kiwi" "xkt" | "cat" "tej" | "xerox" "efm" | "umbrella" "tri"
    | "lighthouse" "plz" | "bellflower" "nqu" | "drum" "bdt" | "dragonfly" "jji"
    | "trampoline" "ohs" | "fireplace" "pbo" | "unicorn" "liz" | "giraffe" "von"
    | "desk" "cec" | "eagle" "ojk" | "yacht" "eww"
```

Generalization Grammar (1/80)

```
S -> NP VP
NP -> DET SUBJ
SUBJ -> "tangerine" | "violin" | "vase" | "whistle" | "sunset" | "yarn" | "ivory"
    | "quasar" | "notebook" | "elephant" | "gazelle" | "harp" | "mosquito" | "xylum"
    | "feather" | "cactus" | "rosemary" | "parrot" | "gelato" | "saxophone" | "coffee"
    | "candle" | "tree" | "nougat" | "rattlesnake" | "guitar" | "jacket" | "zenith"
    | "quiver" | "envelope" | "flower" | "accordion" | "upholstery" | "vortex"
    | "xenon" | "apple" | "banana" | "quilt" | "kettle" | "apricot" | "shovel"
    | "dog" | "penguin" | "sandcastle" | "xylophone" | "ocean" | "muffin"
    | "kangaroo" | "jigsaw" | "puzzle" | "honey" | "wallet" | "lemonade" | "zebra"
    | "lawyer" | "fountain" | "volcano" | "nutmeg" | "mountain" | "rocket"
    | "quill" | "kayak" | "jungle" | "lamp" | "whale" | "puffin" | "airplane"
    | "telescope" | "zeppelin" | "mango" | "kiwi" | "cat" | "xerox" | "umbrella"
    | "lighthouse" | "bellflower" | "drum" | "dragonfly" | "trampoline" | "fireplace"
    | "unicorn" | "giraffe" | "desk" | "eagle" | "yacht"
VP -> VERB DET OBJ | ADV VERB DET OBJ
ADV -> "from time to time"
VERB -> "sees" | "hears" | "resembles" | "likes" | "dislikes"
DET -> "the" | "a"
OBJ -> "tangerine" "DO" | "violin" "DO" | "vase" "DO" | "whistle" "DO" | "sunset" "DO"
```