A Generative Model of Phonotactics

Richard Futrell
Brain and Cognitive Sciences
Massachusetts Institute of Technology
futrell@mit.edu

Adam Albright
Department of Linguistics
Massachusetts Institute of Technology
albright@mit.edu

Peter Graff
Intel Corporation
graffmail@gmail.com

Timothy J. O'Donnell
Department of Linguistics
McGill University
timothy.odonnell@mcgill.ca

Abstract

We present a probabilistic model of phonotactics, the set of well-formed phoneme sequences in a language. Unlike most computational models of phonotactics (Hayes and Wilson, 2008; Goldsmith and Riggle, 2012), we take a fully generative approach, modeling a process where forms are built up out of subparts by phonologically-informed structure building operations. We learn an inventory of subparts by applying stochastic memoization (Johnson et al., 2007; Goodman et al., 2008) to a generative process for phonemes structured as an and-or graph, based on concepts of feature hierarchy from generative phonology (Clements, 1985; Dresher, 2009). Subparts are combined in a way that allows tier-based feature interactions. We evaluate our models' ability to capture phonotactic distributions in the lexicons of 14 languages drawn from the WOLEX corpus (Graff, 2012). Our full model robustly assigns higher probabilities to held-out forms than a sophisticated N-gram model for all languages. We also present novel analyses that probe model behavior in more detail.

1 Introduction

People have systematic intuitions about which sequences of sounds would constitute likely or unlikely words in their language: Although *blick* is not an English word, it sounds like it could be, while *bnick* does not (Chomsky and Halle, 1965). Such intuitions reveal that speakers are aware of the restrictions on sound sequences which can make up possible morphemes in their language—the *phonotactics* of the language. Phonotactic restrictions mean that each language uses only a subset of the logically, or even articulatorily, possible strings of phonemes. Admissible phoneme combinations, on the other hand, typically recur in multiple morphemes, leading to redundancy.

It is widely accepted that phonotactic judgments may be gradient: the nonsense word *blick* is better as a hypothetical English word than *bwick*, which is better than *bnick* (Hayes and Wilson, 2008; Albright, 2009; Daland et al., 2011). To account for such graded judgements, there have been a variety of probabilistic (or, more generally, weighted) models proposed to handle phonotactic learning and generalization over the last two decades (see Daland et al. (2011) and below for review). However, inspired by optimality-theoretic approaches to phonology, the most linguistically informed and successful such models have been *constraint-based*—formulating the problem of phonotactic generalization in terms of restrictions that penalize illicit combinations of sounds (e.g., ruling out $^{*}bn$-).

In this paper, by contrast, we adopt a *generative* approach to modeling phonotactic structure. Our approach harkens back to early work on the sound structure of lexical items which made use of *morpheme structure rules* or *conditions* (Halle, 1959; Stanley, 1967; Booij, 2011; Rasin and Katzir, 2014). Such approaches explicitly attempted to model the

Transactions of the Association for Computational Linguistics, vol. 5, pp. 73–86, 2017. Action Editor: Eric Fosler-Lussier.
Submission batch: 8/2016; Revision batch: 11/2016; Published 2/2017.

redudancy within the set of allowable lexical forms in a language. We adopt a probabilistic version of this idea, conceiving of the phonotactic system as the component of the linguistic system which generates the phonological form of lexical items such as words and morphemes.[1] Our system learns inventories of reusable phonotactically licit structures from existing lexical items, and assembles new lexical items by combining these learned phonotactic patterns using phonologically plausible structure-building operations. Thus, instead of modeling phonotactic generalizations in terms of constraints, we treat the problem as a problem of learning language specific inventories of phonological units and language specific biases on how these phones are likely to be combined.

Although there have been a number of earlier generative models of phonotactic structure (see Section 4) these models have mostly used relatively simplistic or phonologically implausible representations of phones and phonological structure-building. By contrast, our model is built around three representational assumptions inspired by the generative phonology literature. First, we capture sparsity in the space of feature-specifications of phonemes by using *feature dependency graphs*—an idea inspired by work on feature geometries and the contrastive hierarchy (Clements, 1985; Dresher, 2009). Second, our system can represent phonotactic generalizations not only at the level of fully specified segments, but also allows the storage and reuse of *subsegments*, inspired by the autosegments and class nodes of autosegmental phonology. Finally, also inspired by autosegmental phonology, we make use of a structure-building operation which is senstitive to *tier-based contextual structure*.

To model phonotactic learning, we make use of tools from Bayesian nonparametric statistics. In particular, we make use of the notion of *lexical memoization* (**?**; Goodman et al., 2008; Wood et al., 2009; O'Donnell, 2015)—the idea that language-specific generalizations can be captured by the storage and reuse of frequent patterns from a linguistically universal inventory. In our case, this amounts to the idea that an inventory of segments and subsegments can be acquired by a learner that stores and reuses commonly occuring segments in particular, phonologically relevant contexts. In short, we view the problem of learning the phoneme inventory as one of concentrating probability mass on the segments which have been observed before, and the problem of phonotactic generalization as learning which (sub-)segments are likely in particular tier-based phonological contexts.

2 Model Motivations

In this section, we give an overview of how our model works and discuss the phenomena and theoretical ideas that motivate it.

2.1 Feature Dependency Graphs

Most formal models of phonology posit that segments are grouped into sets, known as *natural classes*, that are characterized by shared articulatory and acoustic properties, or *phonological features* (Trubetzkoy, 1939; Jakobson et al., 1952; Chomsky and Halle, 1968). For example, the segments /n/ and /m/ are classified with a positive value of a nasality feature (i.e., NASALITY:+). Similarly, /m/ and /p/ can be classified using the `labial` value of a PLACE feature, PLACE:`labial`. These features allow compact description of many phonotactic generalizations.[2]

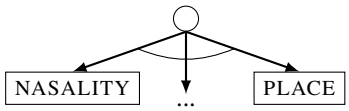
From a probabilistic structure-building perspective, we need to specify a generative procedure which assembles segments out of parts defined in terms of these features. In this section, we will build up such a procedure starting from the simplest possible procedure and progressing towards one which is more phonologically informed. We will clarify the

[1] Ultimately, we conceive of phonotactics as the module of phonology which generates the *underlying* forms of lexical items, which are then subject to phonological transformations (i.e., transductions). In this work, however, we do not attempt to model transformations from underlying to surface forms.

[2] For compatibility with the data sources used in evaluation (Section 5.2), the feature system we use here departs in several ways from standard feature sets: (1) We use multivalent rather than binary-valued features. (2) We represent manner with a single feature, which has values such as `vocalic`, `stop`, and `fricative`. This approach allows us to refer to manners more compactly than in systems that employ combinations of features such as `sonorant`, `continuant`, and `consonantal`. For example, rather than referring to vowels as 'non-syllabic', we refer to them using feature value `vocalic` for the feature MANNER.

generative process here using an analogy to PCFGs, but this analogy will break down in later sections.

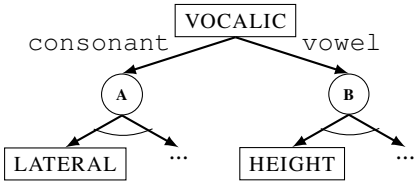
The simplest procedure for generating a segment from features is to specify each feature independently. For example, consider the set of feature-value pairs for /t/: {NASALITY:−, PLACE:alveolar, ...}. In a naive generative procedure, one could generate an instance of /t/ by independently choosing values for each feature in the set {NASALITY, PLACE, ...}. We express this process using the *and-or graph* notation below. Box-shaped nodes—called *or-nodes*—represent features such as NASALITY, while circular nodes represent groups of features whose values are chosen independently and are called *and-nodes*.



This generative procedure is equivalent (ignoring order) to a PCFG with rules:

SEGMENT → NASALITY ... PLACE
NASALITY → +
NASALITY → −
PLACE → bilabial
PLACE → alveolar
...

Not all combinations of feature-value pairs correspond to possible phonemes. For example, while /l/ is distinguished from other consonants by the feature LATERAL, it is incoherent to specify vowels as LATERAL. In order to concentrate probability mass on real segments, our process should optimally assign zero probability mass to these incoherent phonemes. We can avoid specifying a LATERAL feature for vowels by structuring the generative process as below, so that the LATERAL or-node is only reached for consonants:



Beyond generating well-formed phonemes, a basic requirement of a model of phonotactics is that it concentrates mass only on the segments in a particular language's *segment inventory*. For example, the model of English phonotactics should put zero or nominal mass on any sequence containing the segment /x/, although this is a logically possible phoneme. So our generative procedure for a phoneme must be able to learn to generate only the licit segments of a language, given some probability distributions at the and- and or-nodes. For this task, independently sampling values at and-nodes does not give us a way to rule out particular combinations of features such as those forming /x/.

Our approach to this problem uses the idea of *stochastic memoization* (or *adaptation*), in which the results of certain computations are stored and may be probabilistically reused "as wholes," rather than recomputed from scratch (Michie, 1968; Goodman et al., 2008). This technique has been applied to the problem of learning lexical items at various levels of linguistic structure (de Marcken, 1996; Johnson et al., 2007; Goldwater, 2006; O'Donnell, 2015). Given our model so far, applying stochastic memoization is equivalent to specifying an adaptor grammar over the PCFGs described so far.

Let f be a stochastic function which samples feature values using the and-or graph representation described above.We apply stochastic memoization to each node. Following Johnson et al. (2007) and Goodman et al. (2008), we use a distribution for probabilistic memoization known as the Dirichlet Process (DP) (Ferguson, 1973; Sethuraman, 1994). Let $\mathrm{mem}\{f\}$ be a DP-memoized version of f. The behavior of a DP-memoized function can be described as follows. The first time we invoke $\mathrm{mem}\{f\}$, the feature specification of a new segment will be sampled using f. On subsequent invocations, we either choose a value from among the set of previous sampled values (a **memo draw**), or we draw a new value from f (a **base draw**). The probability of sampling the ith old value in a memo draw is $\frac{n_i}{N+\theta}$, where N is the number of tokens sampled so far, n_i is the number of times that value i has been used in the past, and $\theta > 0$ is a parameter of the model. A base draw happens with probability $\frac{\theta}{N+\theta}$. This process induces a bias to reuse items from f which have been frequently generated in the past.

We apply mem recursively to the sampling procedure for each node in the feature dependency graph. The more times that we use some particular set of features under a node to generate words in a language, the more likely we are to reuse that set of

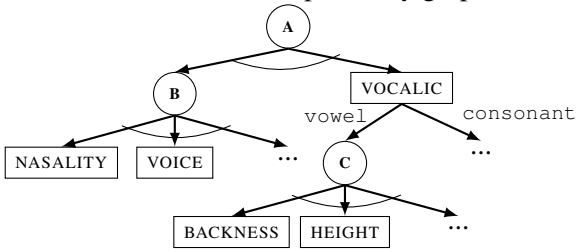
features in the future in a memo draw. This dynamic leads our model to rapidly concentrate probability mass on the subset of segments which occur in the inventory of a language.

2.2 Class Node Structure

Our use of and-or graphs and lexical memoization to model inter-feature dependencies is inspired by work in phonology on distinctiveness and markedness hierarchies (Kean, 1975; Berwick, 1985; Dresher, 2009). In addition to using feature hierarchies to delineate possible segments, the literature has used these structures to designate bundles of features that have privileged status in phonological description, i.e. feature geometries (Clements, 1985; Halle, 1995; McCarthy, 1988). For example, many analyses group features concerning laryngeal states (e.g., VOICE, ASPIRATION, etc.) under a *laryngeal* node, which is distinct from the node containing oral place-of-articulation features (Clements and Hume, 1995). These nodes are known as *class nodes*. In these analyses, features grouped together under the laryngeal class node may covary while being independent of features grouped under the oral class node.

The lexical memoization technique discussed above captures this notion of class node directly, because the model learns an inventory of *subsegments* under each node.

Consider the feature dependency graph below.



In this graph, the and-node **A** generates fully specified segments. And-node **B** can be thought of as generating the non-oral properties of a segment, including voicing and nasality. And-node **C** is a class node bundling together the oral features of vowel segments.

The features under **B** are outside of the VOCALIC node, so these features are specified for both `consonant` and `vowel` segments. This allows combinations such as voiced nasal consonants, and also rarer combinations such as unvoiced nasal vowels. Because all and-nodes are recursively memoized, our model is able to bind together particular non-oral choices (node **B**), learning for instance that the combination {NASALITY:+, VOICED:+} commonly recurs for both vowels and consonants in a language. That is, {NASALITY:+, VOICED:+} becomes a high-probability memo draw.

Since the model learns an inventory of fully specified segments at node **A**, the model could learn one-off exceptions to this generalization as well. For example, it could store at a high level a segment with {NASALITY:+, VOICED:−} along with some other features, while maintaining the generalization that {NASALITY:+, VOICED:+} is highly frequent in base draws. Language-specific phoneme inventories abound with such combinations of class-node-based generalizations and idiosyncrasies. By using lexical memoization at multiple different levels, our model can capture both the broader generalizations described in class node terminology and the exceptions to those generalizations.

2.3 Sequential Structure as Memoization in Context

In Section 2.2, we focused on the role that features play in defining a language's segment inventory. We gave a phonologically-motivated generative process, equivalent to an adaptor grammar, for phonemes in isolation. However, features also play an important role in characterizing licit *sequences*. We model sequential restrictions as *context-dependent segment inventories*. Our model learns a distribution over segments and subsegments conditional on each preceding sequence of (sub)segments, using lexical memoization. Introducing context-dependence means that the model can no longer be formulated as an adaptor grammar.

2.4 Tier-based Interaction

One salient property of sequential restrictions in phonotactics is that segments are often required to bear the same feature values as nearby segments. For example, a sequence of a nasal and a following stop must agree in place features at the end of a morpheme in English. Such restrictions may even be *non-local*. For example, many languages prefer combinations of vowels that agree in features such as HEIGHT, BACKNESS, or ROUNDING, even

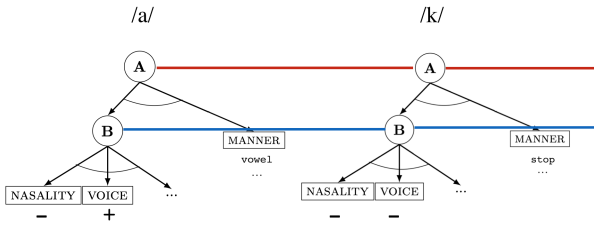


Figure 1: Tiers defined by class nodes **A** and **B** for context sequence /ak/. See text.

across arbitrary numbers of intervening consonants (i.e., *vowel harmony*).

One way to describe these sequential feature interactions is to assume that feature values of one segment in a word depend on values for the same or closely related features in other segments. This is accomplished by dividing segments into subsets (such as consonants and vowels), called *tiers*, and then making a segment's feature values preferentially dependent on the values of other segments on the same tier.

Such phonological tiers are often identified with class nodes in a feature dependency graph. For example, a requirement that one vowel identically match the vowel in the preceding syllable would be stated as a requirement that the vowel's HEIGHT, BACKNESS, and ROUNDING features match the values of the preceding vowel's features. In this case, the vowels themselves need not be adjacent—by assuming that vowel quality features are not present in consonants, it is possible to say that two vowels are adjacent on a tier defined by the nodes HEIGHT, BACKNESS, and ROUNDING.

Our full generative process for a segment following other segments is the following. We follow the example of the generation of a phoneme conditional on a preceding context of /ak/, shown with simplified featural specifications and tiers in Figure 1.

At each node in the feature dependency graph, we can either generate a fully-specified subsegment for that node (memo draw), or assemble a novel subsegment for that node out of parts defined by the feature dependency graph (base draw). Starting at the root node of the feature dependency graph, we decide whether to do a memo draw or base draw conditional on the previous n subsegments at that node.

So in order to generate the next segment following /ak/ in the example, we start at node **A** in the next draw from the feature geometry, with some probability we do a memo draw conditioned on /ak/, defined by the red tier. If we decide to do a base draw instead, we then repeat the procedure conditional on the previous $n-1$ segments, recursively until we are conditioning on the empty context. That is, we do a memo draw conditional on /k/, or conditional on the empty context. This process of conditioning on successively smaller contexts is a standard technique in Bayesian nonparametric language modeling (Teh, 2006; Goldwater et al., 2006).

At the empty context, if we decide to do a base draw, then we generate a novel segment by repeating the whole process at each child node, to generate several subsegments. In the example, we would assemble a phoneme by independently sampling subsegments at the nasal/laryngeal node **B** and the MANNER node, and then combining them. Crucially, the conditioning context consists only of the values *at the current node* in the previous phonemes. So when we sample a subsegment from node **B**, it is conditional on the previous two values at node **B**, { VOICE:+, NASAL:−} and { VOICE:−, NASAL:−}, defined by the blue tier in the figure. The process continues down the feature dependency graph recursively. At the point where the model decides on vowel place features such as height and backness, these will be conditioned only on the vowel places features of the preceding /a/, with /k/ skipped entirely as it does not have values at vowel place nodes.

This section has provided motivations and a walkthrough of our proposed generative procedure for sequences of segments. In the next section, we give the formalization of the model.

3 Formalization of the Models

Here we give a full formal description of our proposed model in three steps. First, in Section 3.1, we formalize the generative process for a segment in isolation. Second, in Section 3.2, we give formulation of Bayesian nonparametric N-gram models with backoff. Third, in Section 3.3, we show how to drop the generative process for a phoneme into the N-gram model such that tier-based interactions emerge naturally.

3.1 Generative Process for a Segment

A feature dependency graph $\mathbf{G}$ is a fully connected, singly rooted, directed, acyclic graph given by the triple $\langle V, A, \mathtt{t}, r\rangle$ where V is a set of vertices or nodes, A is a set of *directed arcs*, $\mathtt{t}$ is a total function $\mathtt{t}(n) : V \mapsto \{\mathtt{and}, \mathtt{or}\}$, and r is a distinguished *root node* in V. A directed arc is a pair $\langle p, c\rangle$ where the *parent* p and *child* c are both elements in V. The function $\mathtt{t}(n)$ identifies whether n is an and- or or-node. Define $\mathtt{ch}(n)$ to be the function that returns all children of node n, that is, all $n' \in N$ such that $\langle n, n'\rangle \in A$.

A *subgraph* $\mathbf{G^s}$ of feature dependency graph $\mathbf{G}$ is the graph obtained by starting from node s by retaining only nodes and arcs reachable by traversing arcs starting from s. A *subsegment* p^s is a subgraph rooted in node s for which each or-node contains exactly one outgoing arc. Subsegments represent sampled phone constituents. A *segment* is a subsegment rooted in r—that is, a fully specified phoneme.

The distribution associated with a subgraph $\mathbf{G^s}$ is given by G^s below. G^s is a distribution over subsegments; the distribution for the full graph G^r is a distribution over fully specified segments. We occasionally overload the notation such that $G^s(p^s)$ will refer to the probability mass function associated with distribution G^s evaluated at the subsegment p^s.

$$H^s \sim \text{DP}(\theta^s, G^s) \tag{1}$$

$$G^s(p^s) = \begin{cases} \prod_{s' \in \mathtt{ch}(s)} H^{s'}(p^{s'}) & \mathtt{t}(s) = \text{AND} \\ \sum_{s' \in \mathtt{ch}(s)} \psi^s_{s'} H^{s'}(p^{s'}) & \mathtt{t}(s) = \text{OR} \end{cases}$$

The first case of the definition covers and-nodes. We assume that the leaves of our feature dependency graph—which represent atomic feature values such as the `laryngeal` value of a PLACE feature—are childless and-nodes.

The second case of the definition covers or-nodes in the graph, where $\psi^s_{s'}$ is the probability associated with choosing outgoing arc $\langle s, s'\rangle$ from parent or-node s to child node s'. Thus, or-nodes define mixture distributions over outgoing arcs. The mixture weights are drawn from a Dirichlet process. In particular, for or-node n in the underlying graph $\mathbf{G}$, the vector of probabilities over outgoing edges is distributed as follows.

$$\vec{\psi^s} \sim \text{DP}(\theta^s, \text{UNIFORM}(|\mathtt{ch}(s)|))$$

Note that in both cases the distribution over child subgraphs is drawn from a Dirichlet process, as below, capturing the notion of subsegmental storage discussed above.

3.2 N-Gram Models with DP-Backoff

Let T be a set of discrete objects (e.g., atomic symbols or structured segments as defined in the preceding sections). Let T^* be the set of all finite-length strings which can be generated by combining elements of T, under concatenation, $\cdot$, including the empty string ϵ. A *context*, $\mathbf{u}$ is any finite string beginning with a special distinguished `start` symbol and ending with some sequence in T^*, that is, $\mathbf{u} \in \{\mathtt{start} \cdot T^*\}$.

For any string α, define $\mathtt{hd}(\alpha)$ to be the function that returns the first symbol in the string, $\mathtt{tl}(\alpha)$ to be the function that returns suffix of α minus the first symbol, and $|\alpha|$ to be the length of α, with $\mathtt{hd}(\epsilon) = \mathtt{tl}(\epsilon) = \epsilon$ and $|\epsilon| = 0$. Write the concatenation of two strings α and α' as $\alpha \cdot \alpha'$.

Let $H_{\mathbf{u}}$ be a distribution on next symbols—that is, objects in $T \cup \{\mathtt{stop}\}$—conditioned on a given context $\mathbf{u}$. For an N-gram model of order N, the probability of a string β in T^* is given by $K^N_{\mathtt{start}}(\beta \cdot \mathtt{stop})$, where $K^N_{\mathbf{u}}(\alpha)$ is defined as:

$$K^N_{\mathbf{u}}(\alpha) = \begin{cases} 1 & \alpha = \epsilon \\ H_{f_N(\mathbf{u})}(\mathtt{hd}(\alpha)) \times K^N_{\mathbf{u}\cdot\mathtt{hd}(\alpha)}(\mathtt{tl}(\alpha)) & \text{otherwise} \end{cases}, \tag{2}$$

where $f_n(\cdot)$ is a context-management function which determines which parts of the left-context should be used to determine the probability of the current symbol. In the case of the N-gram models used in this paper, $f_n(\cdot)$ takes a sequence $\mathbf{u}$ and returns only the rightmost $n-1$ elements from the sequence, or the entire sequence if it has length less than n.

Note two aspects of this formulation of N-gram models. First, $H_{\mathbf{u}}$ is a family of distributions over next symbols or more general objects. Later, we will drop in phonological-feature-based generative processes for these distributions. Second, the function f_n is a parameter of the above definitions. In what follows, we will use a variant of this function which is sensitive to tier-based structure, returning the previous $n-1$ only on the appropriate tier.

MacKay and Peto (1994) introduced a hierarchical Dirichlet process-based backoff scheme for N-

gram models, with generalizations in Teh (2006) and Goldwater et al. (2006). In this setup, the distribution over next symbols given a context $\mathbf{u}$ is drawn hierarchically from a Dirichlet process whose base measure is another Dirichlet process associated with context $\texttt{tl}(\mathbf{u})$, and so on, with all draws ultimately backing off into some unconditioned distribution over all possible next symbols. That is, in a hierarchical Dirichlet process N-gram model, $H_{f_n(\mathbf{u})}$ is given as follows.

$$H_{f_n(\mathbf{u})} \sim \begin{cases} \text{DP}(\theta_{f_n(\mathbf{u})}, H_{f_{n-1}(\mathbf{u})}) & n \geq 1 \\ \text{DP}(\theta_{f_n(\mathbf{u})}, \text{UNIFORM}(T \cup \{\texttt{stop}\})) & n = 0 \end{cases}$$

3.3 Tier-Based Interactions

To make the N-gram model defined in the last section capture tier-based interactions, we make two changes. First, we generalize the generative process H^s from Equation 1 to $H^s_{\mathbf{u}}$, which generates subsegments conditional on a sequence $\mathbf{u}$. Second, we define a context-truncating function $f^s_n(\mathbf{u})$ which takes a context of segments $\mathbf{u}$ and returns the rightmost $n-1$ non-empty subsegments whose root node is s. Then we substitute the generative process $H^s_{f^s_n(\mathbf{u})}$ (which applies the context-management function $f^s_n(\cdot)$ to the context $\mathbf{u}$) for $H_{f_n(\mathbf{u})}$ in Equation 2. The resulting probability distribution is:

$$K^N_{\mathbf{u}}(\alpha) = \begin{cases} 1 & \alpha = \epsilon \\ H^T_{f^s_N(\mathbf{u})}(\texttt{hd}(\alpha)) \times K^N_{\mathbf{u}\cdot\texttt{hd}(\alpha)}(\texttt{tl}(\alpha)) & \text{otherwise} \end{cases}.$$

$K^N_{\mathbf{u}}(\alpha)$ is the distribution over continuations given a context of segments. Its definition depends on $H^s_{f^s_n(\mathbf{u})}$, which is the generalization of the generative process for segments H^s to be conditional on some tier-based N-gram context $f^s_n(\mathbf{u})$. $H^s_{f^s_n(\mathbf{u})}$ is:

$$H^s_{f^s_n(\mathbf{u})} \sim \begin{cases} \text{DP}(\theta^s_{f^s_n(\mathbf{u})}, H^s_{f^s_{n-1}(\mathbf{u})}) & n \geq 1 \\ \text{DP}(\theta^s_{f^s_n(\mathbf{u})}, G^s_{f^s_N(\mathbf{u})}) & n = 0 \end{cases}$$

$$G^s_{f^s_n(\mathbf{u})}(p^s) = \begin{cases} \prod_{s' \in \texttt{ch}(s)} H^{s'}_{f^{s'}_n(\mathbf{u})}(p^{s'}) & \texttt{t}(s) = \text{AND} \\ \sum_{s' \in \texttt{ch}(s)} \psi^s_{s'} H^{s'}_{f^{s'}_n(\mathbf{u})}(p^{s'}) & \texttt{t}(s) = \text{OR}. \end{cases}$$

$H^s_{f^s_n(\mathbf{u})}$ and $G^s_{f^s_n(\mathbf{u})}$ above are mutually recursive functions. $H^s_{f^s_n(\mathbf{u})}$ implements backoff in the tier-based context of previous subsegments; $G^s_{f^s_n(\mathbf{u})}$ implements backoff by going down into the probability distributions defined by the feature dependency graph.

Note that the function $H^s_{f^s_n(\mathbf{u})}$ recursively backs off to the empty context, but its ultimate base distribution is indexed by $f^s_N(\mathbf{u})$, using the global maximum N-gram order N. So when samples are drawn from the feature dependency graph, they are conditioned on non-empty tier-based contexts. In this way, subsegments are generated based on tier-based context and based on featural backoff in an interleaved fashion.

3.4 Inference

We use the Chinese Restaurant Process representation for sampling. Inference in the model is over seating arrangements for observations of subsegments and over the hyperparameters θ for each restaurant. We perform Gibbs sampling on seating arrangements in the Dirichlet N-gram models by removing and re-adding observations in each restaurant. These Gibbs sweeps had negligible impact on model behavior. For the concentration parameter θ, we set a prior $\text{Gamma}(10, .1)$. We draw posterior samples using the slice sampler described in Johnson and Goldwater (2009). We draw one posterior sample of the hyperparameters for each Gibbs sweep. In contrast to the Gibbs sweeps, we found resampling hyperparameters to be crucial for achieving the performance described below (Section 5.3).

4 Related Work

Phonotactics has proven a fruitful problem domain for computational models. Most such work has adopted a constraint-based approach, attempting to design a scoring function based on phonological features to separate acceptable forms from unacceptable ones, typically by formulating restrictions or constraints to rule out less-good structures.

This concept has led naturally to the use of undirected (maximum-entropy, log-linear) models. In this class of models, a form is scored by evaluation against a number of predicates, called *factors*[3]—for example, whether two adjacent segments have the phonological features VOICE:+ VOICE:−. Each factor is associated with a weight, and the score for a form is the sum of the weights of the factors which are true for the form. The well-known model of

[3]Factors are also commonly called "features"—a term we avoid to prevent confusion with phonological features.

Hayes and Wilson (2008) adopts this framework, pairing it with a heuristic procedure for finding explanatory factors while preventing overfitting. Similarly, Albright (2009) assigns a score to forms based on factors defined over natural classes of adjacent segments. Constraint-based models have the advantage of flexibility: it is possible to score forms using arbitrarily complex and overlapping sets of factors. For example, one can state a constraint against adjacent phonemes having features VOICE:+ and LATERAL:+, or any combination of feature values.

In contrast, we have presented a model where forms are built out of parts by structure-building operations. From this perspective, the goal of a model is not to rule out bad forms, but rather to discover repeating structures in good forms, such that new forms with those structures can be generated.

In this setting there is less flexibility in how phonological features can affect well-formedness. For a structure-building model to assign "scores" to arbitrary pairs of co-occurring features, there must be a point in the generative process where those features are considered in isolation. Coming up with such a process has been challenging. As a result of this limitation, structure-building models of phonotactics have not generally included rich featural interactions. For example, Coleman and Pierrehumbert (1997) give a probabilistic model for phonotactics where words are generated using grammar over units such as syllables, onsets, and rhymes. This model does not incorporate fine-grained phonological features such as voicing and place.

In fact, it has been argued that a constraint-based approach is *required* in order to capture rich feature-based interactions. For example, Goldsmith and Riggle (2012) develop a tier-based structure-building model of Finnish phonotactics which captures nonlocal vowel harmony interactions, but argue that this model is inadequate because it does not assign higher probabilities to forms than an N-gram model, a common baseline model for phonotactics (Daland et al., 2011). They argue that this deficiency is because the model cannot simultaneously model nonlocal vowel-vowel interactions and local consonant-vowel interactions. Because of our tier-based conditioning mechanism (Sections 2.4 and 3.3), our model can simultaneously produce local and nonlocal interactions between features using structure-building operations, and does assign higher probabilities to held-out forms than an N-gram model (Section 5.3). From this perspective, our model can be seen as a proof of concept that it is possible to have rich feature-based conditioning without adopting a constraint-based approach.

While our model can capture featural interactions, it is less flexible than a constraint-based model in that the allowable interactions are specified by the feature dependency graph. For example, there is no way to encode a direct constraint against adjacent phonemes having features VOICE:+ and LATERAL:+. We consider this a strength of the approach: A particular feature dependency graph is a parameter of our model, and a specific scientific hypothesis about the space of likely featural interactions between phonemes, similar to feature geometries from classical generative phonology (Clements, 1985; McCarthy, 1988; Halle, 1995).[4]

While probabilistic approaches have mostly taken a constraint-based approach, recent formal language theoretic approaches to phonology have investigated what basic parts and structure building operations are needed to capture realistic feature-based interactions (Heinz et al., 2011; Jardine and Heinz, 2015). We see probabilistic structure-building approaches such as this work as a way to unify the recent formal language theoretic advances in computational phonology with computational phonotactic modeling.

Our model joins other NLP work attempting to do sequence generation where each symbol is generated based on a rich featural representation of previous symbols (Bilmes and Kirchhoff, 2003; Duh and Kirchhoff, 2004), though we focus more on phonology-specific representations. Our and-or graphs are similar to those used in computer vision to represent possible objects (Jin and Geman, 2006).

5 Model Evaluation and Experiments

Here we evaluate some of the design decisions of our model and compare it to a baseline N-gram model and to a widely-used constraint-based model, BLICK. In order to probe model behavior, we also

[4] We do however note that it may be possible to learn feature hierarchies on a language-by-language basis from universal articulatory and acoustic biases, as suggested by Dresher (2009).

present evaluations on artificial data, and a sampling of "representative forms" preferred by one model as compared to another.

Our model consists of structure-building operations over a learned inventory of subsegments. If our model can exploit more repeated structure in phonological forms than the N-gram model or constraint-based models, then it should assign higher probabilities to forms. The log probability of a form under a model corresponds to the description length of that form under the model; if a model assigns a higher log probability to a form, that means the model is capable of compressing the form more than other models. Therefore, we compare models on their ability to assign high probabilities to phonological forms, as in Goldsmith and Riggle (2012).

5.1 Evaluation of Model Components

We are interested in discovering the extent to which each model component described above— feature dependency graphs (Section 2.1), class node structure (Section 2.2), and tier-based conditioning (Section 2.4)— contributes to the ability of the model to explain wordforms.

To evaluate the contribution of feature dependency graphs, we compare our models with a baseline N-gram model, which represents phonemes as atomic units. For this N-gram model, we use a Hierarchical Dirichlet Process with $n = 3$.

To evaluate feature dependency graphs with and without articulated class node structure, we compare models using the graph shown in Figure 3 (the minimal structure required to produce well-formed phonemes) to models with the graph shown in Figure 2, which includes phonologically motivated "class nodes".[5]

To evaluate tier-based conditioning, we compare models with the conditioning described in Sections 2.4 and 3.3 to models where all decisions are conditioned on the full featural specification of the previous $n - 1$ phonemes. This allows us to isolate improvements due to tier-based conditioning beyond improvements from the feature hierarchy.

[5]These feature dependency graphs differ from those in the exposition in Section 2 in that they do not include a MANNER feature; but rather treat `vowel` as a possible value of MANNER.

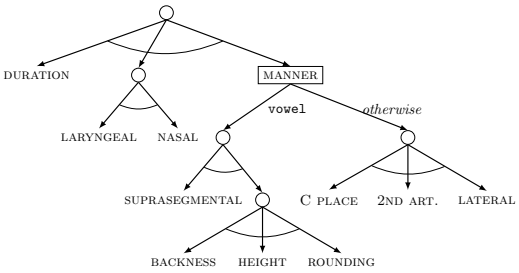


Figure 2: Feature dependency graph with class node structure used in our experiments. Plain text nodes are OR-nodes with no child distributions. The arc marked *otherwise* represents several arcs, each labelled with a consonant manner such as `stop`, `fricative`, etc.

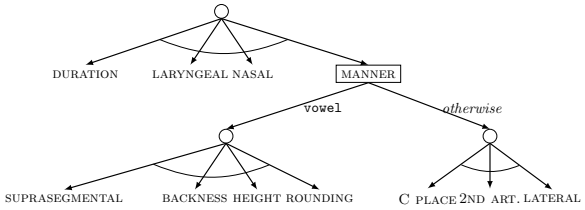


Figure 3: "Flat" feature dependency graph.

5.2 Lexicon Data

The WOLEX corpus provides transcriptions for words in dictionaries of 60 diverse languages, represented in terms of phonological features (Graff, 2012). In addition to words, the dictionaries include some short set phrases, such as *of course*. We use the featural representation of WOLEX, and design our feature dependency graphs to generate only well-formed phonemes according to this feature system. For space reasons, we present the evaluation of our model on 14 of these languages, chosen based on the quality of their transcribed lexicons, and the authors' knowledge of their phonological systems.

5.3 Held-Out Evaluation

Here we test whether the different model configurations described above assign high probability to held-out forms. This tests the models' ability to generalize beyond their training data. We train each model on 2500 randomly selected wordforms from a WOLEX dictionary, and compute posterior predictive probabilities for the remaining wordforms from the final state of the model.

Language	ngram	flat	cl. node	flat/no tiers	cl.node/no tiers
English	-22.20	-22.15	**-21.73****	-22.15	-22.14
French	-18.30	-18.28	**-17.93****	-18.29	-18.28
Georgian	-20.21	-20.17	**-19.64***	-20.18	-20.18
German	-24.77	-24.72	**-24.07****	-24.73	-24.74
Greek	-22.48	-22.45	**-21.65****	-22.45	-22.45
Haitian Creole	-16.09	-16.04	**-15.82****	-16.05	-16.04
Lithuanian	-19.03	-18.99	**-18.58***	-18.99	-18.99
Mandarin	-13.95	-13.83*	**-13.78****	-13.82*	-13.82*
Mor. Arabic	-16.15	-16.10	**-16.00***	-16.13	-16.12
Polish	-20.12	-20.08	**-19.76****	-20.08	-20.07
Quechua	-14.35	-14.30	**-13.87***	-14.30	-14.31
Romanian	-18.71	-18.68	**-18.32****	-18.69	-18.68
Tatar	-16.21	-16.18	**-15.65****	-16.19	-16.19
Turkish	-18.88	-18.85	**-18.55****	-18.85	-18.84

Table 1: Average log posterior predictive probability of a held-out form. "ngram" is the DP Backoff 3-gram model. "flat" models use the feature dependency graph in Figure 3. "cl. node" models use the graph in Figure 2. See text for motivations of these graphs. "no tiers" models condition each decision on the previous phoneme, rather than on tiers of previous features. Asterisks indicate statistical significance according to a t-test comparing with the scores under the N-gram model. * $= p < .05$; ** $= p < .001$.

Table 1 shows the average probability of a held-out word under our models and under the N-gram model for one model run.[6] For all languages, we get a statistically significant increase in probabilities by adopting the autosegmental model with class nodes and tier-based conditioning. Model variants without either component do not significantly outperform the N-gram model except in Chinese. The combination of class nodes and tier-based conditioning results in model improvements beyond the contributions of the individual features.

5.4 Evaluation on Artificial Data

Our model outperforms the N-gram model in predicting held-out forms, but it remains to be shown that this performance is due to capturing the kinds of linguistic intuitions discussed in Section 2. An alternative possibility is that the Autosegmental N-gram model, which has many more parameters than a plain N-gram model, can simply learn a more accurate model of any sequence, even if that sequence has none of the structure discussed above. To evaluate this possibility, we compare the performance of our model in predicting held-out linguistic forms to its performance in predicting held-out forms from artificial lexicons which expressly do *not* have the linguistic structure we are interested in.

If the autosegmental model outperforms the N-gram model even on artificial data with no phonological structure, then its performance on the real linguistic data in Section 5.3 might be overfitting. On the other hand, if the autosegmental model does better on real data but not artificial data, then we can conclude that it is picking up on some real distinctive structure of that data.

For each real lexicon L_r, we generate an artificial lexicon L_a by training a DP 3-gram model on L_r and forward-sampling $|L_r|$ forms. Additionally, the forms in L_a are constrained to have the same distribution over lengths as the forms in L_r. The resulting lexicons have no tier-based or featural interactions except as they appear by chance from the N-gram model trained on these lexica. For each L_a we then train our models on the first 2500 forms and score the probabilities of the held-out forms, the same procedure as in Section 5.3.

We ran this procedure for all the lexicons shown in Table 1. For all but one lexicon, we find that the autosegmental models do not significantly outperform the N-gram models on artificial data. The exception is Mandarin Chinese, where the average log probability of an artificial form is -13.81 under the N-gram model and -13.71 under the full autosegmental model. The result suggests that the anomalous behavior of Mandarin Chinese in Section 5.3 may be due to overfitting.

When exposed to data that explicitly does *not* have autosegmental structure, the model is not more accurate than a plain sequence model for almost all languages. But when exposed to real linguistic data, the model is more accurate. This result provides evidence that the generative model developed in Section 2 captures true distributional properties of lexicons that are absent in N-gram distributions, such as featural and tier-based interactions.

5.5 Comparison with a Constraint-Based Model

Here we provide a comparison with Hayes and Wilson (2008)'s Phonotactic Learner, which outputs a phonotactic grammar in the form of a set of weighted constraints on feature co-occurrences. This grammar is optimized to match the constraint violation profile in a training lexicon, and so can be

[6]The mean standard deviation per form of log probabilities over 50 runs of the full model ranged from .09 for Amharic to .23 for Dutch.

seen as a probabilistic model of that lexicon. The authors have distributed one such grammar, BLICK, as a "reference point for phonotactic probability in experimentation" (Hayes, 2012). Here we compare our model against BLICK on its ability to assign probabilities to forms, as in Section 5.3.

Ideally, we would simply compute the probability of forms like we did in our earlier model comparisons. BLICK returns scores for each form. However, since the probabilistic model underlying BLICK is undirected, these scores are in fact unnormalized log probabilities, so they cannot be compared directly to the normalized probabilities assigned by the other models. Furthermore, because the probabilistic model underlying BLICK does not penalize forms for length, the normalizing constant over all forms is in fact infinite, making straightforward comparison of predictive probabilities impossible. Nevertheless, we can turn BLICK scores into probabilities by conditioning on further constraints, such as the length k of the form. We enumerate all possible forms of length k to compute the normalizing constant for the distribution over forms of that length. The same procedure can also be used to compute the probabilities of each form, conditioned on the length of the form k, under the N-gram and Autosegmental models.

To compare our models against BLICK, we calculate conditional probabilities for forms of length 2 through 5 from the English lexicon.[7] The forms are those in the WOLEX corpus; we include them for this evaluation if they are k symbols long in the WOLEX representation. For our N-gram and Autosegmental models, we use the same models as in Section 5.3. The average probabilities of forms under the three models are shown in Table 2. For length 3-5, the autosegmental model assigns the highest probabilities, followed by the N-gram model and BLICK. For length 2, BLICK outperforms the DP N-gram model but not the autosegmental model.

Our model assigns higher probabilities to short forms than BLICK. That is, our models have identified more redundant structure in the forms than BLICK, allowing them to compress the data more. However, the comparison is imperfect in several ways. First, BLICK and our models were trained on different data; it is possible that our training data are more representative of our test data than BLICK's training data were. Second, BLICK uses a different underlying featural decomposition than our models; it is possible that our feature system is more accurate. Nevertheless, these results show that our model concentrates more probability mass on (short) forms attested in a language, whereas BLICK likely spreads its probability mass more evenly over the space of all possible (short) strings.

[7] Enumerating and scoring the 22,164,361,129 possible forms of length 6 was computationally impractical.

Length	BLICK	ngram	cl. node
2	-6.50	-6.81	**-5.18**
3	-9.38	-8.76	**-7.95**
4	-14.1	-11.7	**-11.4**
5	-18.1	-14.2	**-13.9**

Table 2: Average log posterior predictive probability of an English form of fixed length under BLICK and our models.

English N-gram	English Full Model
collaborationist	mistrustful
a posteriori	inharmoniousness
sacristy	absentmindedness
matter of course	blamelessness
earnest money	phlegmatically

Table 3: Most representative forms for the N-gram model and for our full model ("cl. node" in Table 1) in English. Forms are presented in native orthography, but were scored based on their phonetic form.

5.6 Representative Forms

In order to get a sense of the differences between models, we investigate what phonological forms are preferred by different kinds of models. These forms might be informative about the phonotactic patterns that our model is capturing which are not well-represented in simpler models. We calculate the *representativeness* of a form f with respect to model m_1 as opposed to m_2 as $p(f|m_1)/p(f|m_2)$ (Good, 1965; Tenenbaum and Griffiths, 2001). The forms that are most "representative" of model m_1 are not the forms that m_1 assigns the highest probability, but rather the forms that m_1 ranks highest relative to m_2.

Tables 3 and 4 show forms from the lexicon that are most representative of our full model and of the N-gram model for English and Turkish. The most

Turkish N-gram	Turkish Full Model
üstfamilya	büyükkarapınar
dekstrin	kızılcapınar
mnemotekni	altınpınar
ekskavatör	sarımehmetler
foksterye	karaelliler

Table 4: Most representative forms for N-gram and Autosegmental models in Turkish.

uniquely representative forms for our full model are morphologically complex forms consisting of many productive, frequently reused morphemes such as *ness*. On the other hand, the representative forms for the N-gram model include foreign forms such as *a posteriori* (for English) and *ekskavatör* (for Turkish), which are not built out of parts that frequently repeat in those languages. The representative forms suggest that the full model places more probability mass on words which are built out of highly productive, phonotactically well-formed parts.

6 Discussion

We find that our models succeed in assigning high probabilities to unseen forms, that they do so specifically for linguistic forms and not random sequences, that they tend to favor forms with many productive parts, and that they perform comparably to a state-of-the-art constraint-based model in assigning probabilities to short forms.

The improvement for our models over the N-gram baseline is consistent but not large. We attribute this to the way in which phonological generalizations are used in the present model: in particular, phonological generalizations function primarily as a form of backoff for a sequence model. Our models have lexical memoization at each node in a feature dependency graph; as such, the *top* node in the graph ends up representing transition probabilities for whole phonemes conditioned on previous phonemes, and the rest of the feature dependency graph functions as a backoff distribution. When a model has been exposed to many training forms, its behavior will be largely dominated by the N-gram-like behavior of the top node. In future work it might be effective to learn an optimal backoff procedure which gives more influence to the base distribution (Duh and Kirchhoff, 2004; Wood and Teh, 2009).

While the tier-based conditioning in our model would seem to be capable of modeling nonlocal interactions such as vowel harmony, we have not found that the models do well at reproducing these nonlocal interactions. We believe this is because the model's behavior is dominated by nodes high in the feature dependency graph. In any case, a simple Markov model defined over tiers, as we have presented here, might not be enough to fully model vowel harmony. Rather, a model of phonological *processes*, transducing underlying forms to surface forms, seems like a more natural way to capture these phenomena.

We stress that this model is not tied to a particular feature dependency graph. In fact, we believe our model provides a novel way of testing different hypotheses about feature structures, and could form the basis for learning the optimal feature hierarchy for a given data set. The choice of feature dependency graph has a large effect on what featural interactions the model can represent directly. For example, neither feature dependency graph has shared place features for consonants and vowels, so the model has limited ability to represent place-based restrictions on consonant-vowel sequences such as requirements for labialized or palatalized consonants in the context of /u/ or /i/. These interactions can be treated in our framework if vowels and consonants share place features, as in Padgett (2011).

7 Conclusion

We have presented a probabilistic generative model for sequences of phonemes defined in terms of phonological features, based on representational ideas from generative phonology and tools from Bayesian nonparametric modeling. We consider our model as a proof of concept that probabilistic structure-building models can include rich featural interactions. Our model robustly outperforms an N-gram model on simple metrics, and learns to generate forms consisting of highly productive parts. We also view this work as a test of the scientific hypotheses that phonological features can be organized in a hierarchy and that they interact along tiers: in our model evaluation, we found that both concepts were necessary to get an improvement over a baseline N-gram model.

Acknowledgments

We would like to thank Tal Linzen, Leon Bergen, Edward Flemming, Edward Gibson, Bob Berwick, Jim Glass, and the audiences at MIT's Phonology Circle, SIGMORPHON, and the LSA 2016 Annual Meeting for helpful comments. This work was supported in part by NSF DDRIG Grant #1551543 to R.F.

References

Adam Albright. 2009. Feature-based generalization as a source of gradient acceptability. *Phonology*, 26:9–41.

Robert C. Berwick. 1985. *The acquisition of syntactic knowledge.* MIT Press, Cambridge, MA.

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003–Short Papers–Volume 2*, pages 4–6. Association for Computational Linguistics.

Geert Booij. 2011. Morpheme structure constraints. In *The Blackwell Companion to Phonology*. Blackwell.

Noam Chomsky and Morris Halle. 1965. Some controversial questions in phonological theory. *Journal of Linguistics*, 1:97–138.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row, New York, NY.

George N. Clements and Elizabeth V. Hume. 1995. The internal organization of speech sounds. In *The Handbook of Phonological Theory*, pages 24–306. Blackwell, Oxford.

George N. Clements. 1985. The geometry of phonological features. *Phonology Yearbook*, 2:225–252.

John Coleman and Janet B. Pierrehumbert. 1997. Stochastic phonological grammars and acceptability. In John Coleman, editor, *Proceedings of the 3rd Meeting of the ACL Special Interest Group in Computational Phonology*, pages 49–56, Somserset, NJ. Association for Computational Linguistics.

Robert Daland, Bruce Hayes, James White, Marc Garellek, Andreas Davis, and Ingrid Normann. 2011. Explaining sonority projection effects. *Phonology*, 28:197–234.

Carl de Marcken. 1996. Linguistic structure as composition and perturbation. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 335–341. Association for Computational Linguistics.

B. Elan Dresher. 2009. *The Contrastive Hierarchy in Phonology*. Cambridge University Press.

Kevin Duh and Katrin Kirchhoff. 2004. Automatic learning of language model structure. In *Proceedings of COLING 2004*, Geneva, Switzerland.

Thomas S. Ferguson. 1973. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230.

John Goldsmith and Jason Riggle. 2012. Information theoretic approaches to phonology: the case of Finnish vowel harmony. *Natural langauge and linguistic theory*, 30(3):859–96.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems*, volume 18, pages 459–466, Cambridge, MA. MIT Press.

Sharon Goldwater. 2006. *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Brown University.

Irving John Good. 1965. *The Estimation of Probabilities*. MIT Press, Cambridge, MA.

Noah D. Goodman, Vikash K. Mansinghka, Daniel Roy, Keith Bonawitz, and Joshua B. Tenenbaum. 2008. Church: A language for generative models. In *Uncertainty in Artificial Intelligence*, Helsinki, Finland. AUAI Press.

Peter Graff. 2012. *Communicative Efficiency in the Lexicon*. Ph.D. thesis, Massachusetts Institute of Technology.

Morris Halle. 1959. *The Sound Pattern of Russian: A linguistic and acoustical investigation*. Mouton, The Hague, The Netherlands.

Morris Halle. 1995. Feature geometry and feature spreading. *Linguistic Inquiry*, 26(1):1–46.

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

Bruce Hayes. 2012. Blick - a phonotactic probability calculator.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *The 49th Annual Meeting of the Association for Computational Linguistics*.

Roman Jakobson, C. Gunnar M. Fant, and Morris Halle. 1952. *Preliminaries to Speech Analysis: The Distinctive Features and their Correlates*. The MIT Press, Cambridge, Massachusetts and London, England.

Adam Jardine and Jeffrey Heinz. 2015. A concatenation operation to derive autosegmental graphs. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 139–151, Chicago, USA, July.

Ya Jin and Stuart Geman. 2006. Context and hierarchy in a probabilistic image model. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVRP'06)*, pages 2145–2152.

Mark Johnson and Sharon Goldwater. 2009. Improving nonparameteric bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *NAACL-HLT 2009*, pages 317–325. Association for Computational Linguistics.

Mark Johnson, Thomas L Griffiths, and Sharon Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems*, pages 641–648.

Mary-Louise Kean. 1975. *The Theory of Markedness in Generative Grammar*. Ph.D. thesis, Massachusetts Institute of Technology.

David J.C. MacKay and Linda C. Bauman Peto. 1994. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1:1–19.

John McCarthy. 1988. Feature geometry and dependency: A review. *Phonetica*, 43:84–108.

Donald Michie. 1968. "memo" functions and machine learning. *Nature*, 218:19–22.

Timothy J. O'Donnell. 2015. *Productivity and Reuse in Language: A Theory of Linguistic Computation and Storage*. The MIT Press, Cambridge, Massachusetts and London, England.

Jaye Padgett. 2011. Consonant-vowel place feature interactions. In *The Blackwell Companion to Phonology*, pages 1761–1786. Blackwell Publishing, Malden, MA.

Ezer Rasin and Roni Katzir. 2014. A learnability argument for constraints on underlying representations. In *Proceedings of the 45th Annual Meeting of the North East Linguistic Society (NELS 45)*, Cambridge, Massachusetts.

Jayaram Sethuraman. 1994. A constructive definition of Dirichlet priors. *Statistica Sinica*, pages 639–650.

Richard Stanley. 1967. Redundancy rules in phonology. *Language*, 43(2):393–436.

Yee Whye Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore.

Joshua B Tenenbaum and Thomas L Griffiths. 2001. The rational basis of representativeness. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 1036–1041.

Nikolai S. Trubetzkoy. 1939. *Grundzüge der Phonologie*. Number 7 in Travaux du Cercle Linguistique de Prague. Vandenhoeck & Ruprecht, Göttingen.

Frank Wood and Yee Whye Teh. 2009. A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation. In *Artificial Intelligence and Statistics*, pages 607–614.

Frank Wood, Cédric Archambeau, Jan Gasthaus, James Lancelot, and Yee Whye Teh. 2009. A stochastic memoizer for sequence data. In *Proceedings of the 26 th International Conference on Machine Learning*, Montreal, Canada.