Learning rule-based morpho-phonology

Ezer Rasin, Iddo Berger, Nur Lan, and Roni Katzir

June 19, 2018

1 Introduction

As part of language acquisition, the child needs to acquire many different aspects of the morpho-phonology of their language. If the child is learning English, for example, they will need to learn that in 'cats', pronounced $[\mathrm{k}^h\mathrm{æts}]$, the aspiration of the initial $[\mathrm{k}]$ and the voicelessness of the final $[\mathrm{s}]$ are no accident: in English, voiceless stops such as $[\mathrm{k}]$ are always aspirated in this position (roughly, syllable-initially in a stressed syllable), and the expression of the plural morpheme is always the voiceless $[\mathrm{s}]$ after a voiceless stop such as $[\mathrm{t}]$. Thus, the child will need to learn that imaginable forms such as $[\mathrm{kæts}]$ or $[\mathrm{k}^h\mathrm{ætz}]$ are not possible in the language. These pieces of knowledge come from a very large – possibly unbounded – set of possible choices that languages can make and that children must be able to acquire. Moreover, the child acquires much of this knowledge from distributional cues alone, without access to analyzed forms or paradigms and without negative evidence. The result is a nontrivial learning task that is challenging even in relatively simple cases such as deterministic, surface-true phonotactics (as in the aspiration pattern of English) or alternations providing useful information (such as the voicing pattern concerning the $/\mathrm{z}/$ suffix in English). The learning challenge is even more pronounced in cases of optional phonological processes and of opaque interactions of phonological processes. To date, no general solution to this challenge has been provided in the literature.

In this paper we propose a response to the learning challenge in terms of a certain kind of simplicity metric. The simplicity metric will follow the principle of Minimum Description Length (MDL; Rissanen 1978), which incorporates both the idea of grammar simplicity (as in the evaluation metric of early generative phonology) and that of restrictiveness (or how easy it is for the grammar to capture the data). The representational framework that we use for our discussion will be that of rule-based phonology, which offers a particularly direct handle on the representation of both optionality and opacity. In order to illustrate how the MDL metric can guide the learner toward appropriate hypotheses, we present several simulations that start with a small corpus of unanalyzed surface forms – generated from artificial grammars based on morpho-phonological patterns in various languages – and arrive at a full grammar including a lexicon of underlying forms, a morphological segmentation of forms into morphemes and their attachment possibilities, and different kinds of phonological rules (both obligatory and optional) and their ordering (including both transparent and opaque interac-

tions). While it might seem that these different aspects of morpho-phonological knowledge call for a fragmented learning approach, with specialized learners for the different sub-tasks, we will show how the MDL evaluation metric allows all of them to be acquired in a unified way.

We start, in section 2, by presenting the MDL metric in the context of rule-based phonology and by specifying a concrete set of representations for phonological grammars and their MDL costs. In section 3 we present proof-of-concept learning simulations with optionality, rule interaction (including opacity), and interdependent phonology and morphology. To keep the presentation simple, the discussion in the first part of the paper sets aside a variety of proposals in the literature and focuses entirely on two kinds of learning biases – grammar simplicity and restrictiveness – and their combination within the MDL metric. Section 4 discusses previous work on the learning of rule-based morpho-phonology more broadly. Section 5 concludes.

2 The present work

The current section presents the assumptions behind our learning model. We start, in section 2.1, by considering two evaluation metrics from the literature – the evaluation metric of the *Sound Pattern of English* (SPE; Chomsky and Halle 1968, p. 334), which aims for grammar economy, and the subset principle, which aims for restrictiveness – in the context of acquiring a single optional phonological rule. We will see that in order to acquire the relevant rule, the child cannot follow grammar economy alone or restrictiveness alone but must instead balance between the two. This balancing of economy and restrictiveness is the essence of the MDL evaluation metric, and while we motivate it here using one simple rule, the very same metric will allow us to learn whole (though at present artificial) phonological grammars, including the lexicon, the morphological segmentation of forms into stems and affixes, a variety of phonological rules, and both transparent and opaque rule interactions. In order to turn the MDL evaluation metric into an actual phonological learner, we need to adopt explicit representations for phonological grammars. We do this in section 2.2, where we present the concrete representations we assume and the costs they induce in terms of MDL. Section 2.3 completes the description of the learner by presenting the search procedure that we use to find a grammar that yields a good MDL score.

2.1 The MDL criterion

French has an optional process of liquid-deletion word-finally following an obstruent (Dell, 1981). The French-learning child, then, will be exposed to surface forms such as [tabl] and [tab] for 'table' and [katr] and [kat] for 'four' (but only [gar] and not *[ga] for 'train station', since its liquid does not appear in the right environment for deletion). Suppose that the child uses a simplicity metric such as the one in SPE, which optimizes grammar economy:[1]

[1]Here and below the grammar *G* will be taken to be not just the phonological rules and their ordering but also the lexicon. Thus, by saying that a grammar *G* generates the data *D*, we mean that every string in *D* can be derived as a licit surface form from some UR in the lexicon and the ordered phonological rules.

(1) SPE EVALUATION METRIC: If G and G' can both generate the data D, and if $|G| < |G'|$, prefer G to G'

We use $|\cdot|$ to notate length, and to see how we can use (1) we need to be precise about how $|\cdot|$ is measured. Anticipating our discussion below, it will be convenient to think of grammars as sitting in computer memory according to a given encoding scheme, with $|G|$ the number of bits taken up by G. In section 2.2 we will present the details of one specific encoding scheme and show how $|G|$ is measured within it. For now, however, we will set aside such details as we build toward the MDL criterion.

Early on, the child will store a separate UR for each surface form of the alternating pairs: both /tabl/ and /tab/ for 'table'; both /katr/ and /kat/ for 'four'; both /arbr/ and /arb/ for 'tree'; and so on (along with a single /gar/ for 'train station'). After seeing a few additional alternating pairs of this kind, however, (1) will lead the child to conclude that for each such pair there is just one UR – /tabl/ for 'table', /katr/ for 'four', /arbr/ for 'tree', and so on – and that an optional phonological rule such as the following applies (where *L* stands for *liquid*):

(2) $L \rightarrow \emptyset$ (optional)

The rule in (2) adds complexity to the grammar, but this complexity is more than offset by the savings obtained by the elimination of all the *L*-less forms from the lexicon. Consequently, the overall size of the grammar is shorter using (2), and (1) will favor the new grammar.

As mentioned above, however, the actual process of *L*-deletion in French is somewhat more specific than (2) suggests: *L* may be deleted, but only in certain contexts. A more appropriate rule is the following, in which *L*-deletion is restricted to word-final environments following an obstruent:

(3) $L \rightarrow \emptyset$ /[$-$*son*]__# (optional)

And unfortunately, as pointed out by Dell (1981), a child using (1) will fail to acquire the appropriate context for the application of the rule. That is, the child will prefer (2) to the more appropriate (3). This is so since (a) both a grammar G using the unrestricted (2) and a grammar G' using the restricted (3) can generate the data; and (b) G is shorter than G' (since specifying the context in (3) adds to the grammar's length). By the SPE evaluation metric in (1), the child will prefer G to G', which is the wrong result. For example, a child using G will erroneously rule in *L*-deleted forms such as *[ga] for /gar/.[2] Moreover, the child will never recover from this error: since the child sees only positive evidence, they will never be forced to leave the simpler but overly inclusive G.

The problem is quite general, as discussed by Braine (1971) and Baker (1979), and goes well beyond phonology: a child guided solely by a preference for grammar

[2]In fact, a preference for grammar economy will lead the learner to even more extreme solutions if left unchecked. In particular, consider a grammar that has an optional epenthesis rule for each segment that appears in the data and a lexicon that consists only of the empty string. Such a grammar can generate the data and is extremely short to state. Unless it is blocked by some other principle, this grammar will be preferred by (1) to both G and G'.

economy, as in the SPE evaluation metric in (1), will fail to learn the contexts for optional rules. Just as in the example of optional *L*-deletion, a grammar *G* in which an optional rule *R* has no context will generally be both simpler and more inclusive than a minimal variant G' in which the optional rule does have a context. If G' is the correct grammar, both grammars will be able to generate the input data: G' since it is the correct grammar, and *G* since its *language* – that is, the set of all licit forms according to the lexicon and rules of *G* – is a superset of the language of G'. By (1), then, the child will incorrectly prefer the simpler *G* to G' and – since the child will not receive negative evidence – will never recover from this error.

One solution to this predicament – the one advocated by Dell (1981) and adopted in much later work – is to change the evaluation metric from one that favors simple grammars to one that favors restrictive ones, where restrictiveness is captured in terms of subsethood: *G* is more restrictive than G' if its language is a subset of the language of G'.[3] This solution, also known as the *subset principle* (Berwick 1985; Wexler and Manzini 1987), directs the learner to never choose a superset language when a proper subset is compatible with the data:[4,5]

(4) SUBSET EVALUATION METRIC: If *G* and G' can both generate the data *D*, and if the language of *G* is a proper subset of the language of G', prefer *G* to G'

A child following (4) will always choose a minimal language compatible with the data and will thus avoid the overgeneralization problem. In the case of optional *L*-deletion in French, the grammar with the unrestricted (2) generates a language that is a strict superset of the one with the restricted (3), and both grammars generate the data *D*; consequently, the unrestricted (2) will be rejected and the restricted (2) chosen, which is the correct result.

While choosing correctly between (2) and (3), the subset principle gives rise to a problem of undergeneralization – the mirror image of the overgeneralization problem of the SPE simplicity metric – and does not offer a general solution for learning. To see the problem in the case of French *L*-deletion, consider the situation of a learner who has heard a surface form such as [sabl] but, accidentally, has not yet heard its *L*-elided variant [sab] (both for the UR /sabl/ 'sand'). If the learner has heard sufficiently many

[3]Other ways of cashing out the informal idea of restrictiveness have been proposed in the literature. Within Optimality Theory (Prince and Smolensky 1993), for example, restrictiveness is often interpreted as subsethood not of the languages of the original grammars *G* and G' but rather of the languages of variants of *G* and G' in which the lexicon is replaced with the set $\Sigma*$ of all possible strings over the alphabet Σ in which the lexicon is written (see Smolensky 1996). The MDL metric, which we will present and argue for below, implements restrictiveness in yet another way, by comparing how easy it is to specify the actual input data using *G* and G': if the data can be more easily specified using *G* than using G', then *G* is the more restrictive grammar of the two.

[4]As Baker (1979) notes, Braine (1971)'s alternative to the SPE evaluation metric, while stated in procedural terms, has a similar effect to a restrictiveness metric.

[5]In the general case, determining whether the language of one (unrestricted, or even just context-free) grammar is a subset of the language of another grammar is undecidable. In the case of phonology, however, where the languages are often assumed to be regular (see Johnson 1972 and Kaplan and Kay 1994, among others), subsethood can be established mechanically in terms of the finite-state automata that generate the two languages. In particular, to check whether L_1 is a subset of $L2$, we can check whether $L_1 \setminus L_2 = \emptyset$. If L_1 and L_2 are both regular, then $L_1 \setminus L_2$ ($= L_1 \cap L_2^c$) is regular (and easily computable), and determining whether a regular language is empty is decidable (see, e.g., Hopcroft et al. 2007).

other pairs differing only in whether they have a final liquid, we would expect them to adopt (3), even if for /sabl/ only one member of the pair has been observed so far. That is, we would like the learner to generalize beyond the data in this case. But if the learner is following the subset principle, this will not be possible: with (3), the language will include also the *L*-deleted form [sab], which makes the language a strict superset of the language of a grammar that does not generate [sab] (for example, a grammar without any deletion rules and with a lexicon that has separate URs for each of the *L*-variants that have been seen in the input data). In other words, a single accidental gap is enough to prevent a learner following the subset principle from making what seems like a reasonable generalization.

We have seen that minimizing |*G*|, as in the SPE evaluation metric, makes the child generalize; when left unchecked, however, it leads to overgeneralization. Meanwhile, restrictiveness (as in the subset principle) protects from overgeneralization, but on its own prevents useful generalizations. It seems sensible, then, to try to balance the two principles against each other: look for a grammar that is both reasonably small and reasonably restrictive. This is exactly the idea behind Minimal Description Length (MDL; Rissanen 1978), which we will adopt here.[6] To make it work, however, we need to specify how we quantify both grammar size and restrictiveness and how the two are balanced. The insight of MDL – building on the work of Solomonoff (1964), Kolmogorov (1965), and Chaitin (1966) – is that we can think of restrictiveness as another simplicity criterion and combine it naturally with grammar economy. As above, for grammar economy we will consider *G* as sitting in computer memory according to a given encoding and measure |*G*| in terms of how many bits the storage of *G* takes up. Restrictiveness, meanwhile, will be thought of in terms of how simple it is to tell the story of the data, *D*, given the grammar, *G*, a story that we will notate as $D : G$.[7] Consider again the case of optional *L*-deletion. Suppose that the learner has acquired a lexicon with the single UR /tabl/ and an optional rule such as (2) or (3). To describe an instance of the surface form [tabl] or the surface form [tab], we need to first specify the UR /tabl/ and then specify whether *L*-deletion has applied (for [tab]) or not (for [tabl]). Specifying the UR /tabl/ involves a choice from among the URs. In general, the greater the number of URs from which we choose, the longer the specification of the UR we have selected. A convenient way of specifying such choices – and one that will allow us to directly balance the length of $D : G$ against that of the grammar *G* – is using bits. A single bit encodes one binary choice, and as the number of bits grows, the number of choices that can be stated grows (exponentially) with it. For example, if there are just two possible URs, we can specify the choice using one bit. With four URs in the lexicon, we now need about two bits to specify each choice.[8] And so on.

[6] See also the closely related idea of Minimal Message Length of Wallace and Boulton (1968).

[7] In what follows, we will consider *D* to be the actual data sequence that the learner is exposed to. Consequently, $D : G$ will be the story of those actual input tokens given the grammar. This choice is made for concreteness and in order to keep the presentation simple. A different possibility would be to abstract away from individual tokens and consider only the types – that is, the distinct surface forms – rather than the tokens. It is also possible to define the restrictiveness factor $|D : G|$ in terms of a combined measure of types and tokens. We will not attempt to investigate these choices and their implications for learning within this paper (see Goldwater et al. 2006, Endress and Hauser 2011, and Yang 2016 for relevant discussion).

[8] Exactly how many bits are needed for each choice will depend on the specific grammar *G*, relative to which the choices are made. In section 2.2 we show how $D : G$ is stated relative to the grammars presented

The optional *L*-deletion rule requires the further specification of whether it applied or not, which can be stated as one additional bit (perhaps 0 to specify that the rule did not apply and 1 to specify that it did). These specifications for the different surface forms in the input data *D* are accumulated to provide the complete $D : G$, the encoding of the specific input data *D* given the grammar *G*.

We can now see how the motivation for restricting the context for optional *L*-deletion can be stated in terms of simplicity. If *L*-deletion were not optional – if it always applied or if it never applied – the final bit would have been unnecessary for the specification of the relevant surface forms: selecting a UR would have fully determined the surface form. For URs like /tabl/ and /katr/, *L*-deletion is optional, and the extra bit of the appropriate rule cannot be avoided. But for /gar/ *L*-deletion never applies, so paying an extra bit for each occurrence is an unnecessary expense. The unrestricted (2) forces us to pay this unnecessary expense: the optional rule is applicable whenever a UR is chosen that contains liquids (and for each occurrence of a liquid within such a UR), including URs such as /gar/ that do not allow for *L*-deletion, so a bit specifying whether the rule applies is always required, leading to $D : G$ that is longer than needed. The more restricted (3), on the other hand, makes us pay the extra bit only when an appropriate UR such as /tabl/ is chosen but not when /gar/ is chosen. Consequently, (3) leads to a shorter $D : G$.

Having recast the notion of restrictiveness in terms of simplicity (specifically, the simplicity of the story of $D : G$), we can immediately see how we can combine this idea with simplicity of grammar: instead of minimizing $|G|$ alone, as in the SPE evaluation metric, we can now minimize the sum of the two quantities, $|G|+|D : G|$, thus balancing between the goal of a simple, general grammar and a restrictive one.

(5) MDL EVALUATION METRIC: If *G* and *G′* can both generate the data *D*, and if $|G| + |D : G| < |G'| + |D : G'|$, prefer *G* to *G′*

In our *L*-deletion example, storing a single UR for pairs like [tabl]/[tab] and [katr]/[kat] will shorten $|G|$ sufficiently (given a large enough number of such pairs) to justify adding an optional rule of *L*-deletion to *G*, just as with the SPE evaluation metric. As for the precise form of the rule, the simultaneous consideration of both $|G|$ and $|D : G|$, as in (5), will mean that the more complex rule in (3) will eventually be chosen over the unrestricted (2), despite its increased $|G|$. The reason is that after sufficiently many instances of [gar] have been encountered, the savings in terms of $|D : G|$ obtained with (3) – since no bit will need to be spent when a UR such as /gar/ is chosen – will more than outweigh the increase in $|G|$. Figure 1 illustrates. The MDL metric in (5) thus allows the child to generalize but protects them from overgeneralizing.

Note that, differently from the case of restrictiveness-only (as in the subset principle), the MDL metric has the means to generalize beyond the data even in the face of certain gaps in the input. Consider again the situation of a learner who has heard the form [sabl] but has not (yet) heard its *L*-deleted variant [sab]. We saw earlier how this kind of gap in the input data will prevent a restrictiveness-only learner from generalizing correctly. For an MDL learner, the added restrictiveness of ruling out [sab]

in that section. For similar considerations regarding the measurement of $|G|$ and $|D : G|$ in bits but within constraint-based phonology see Rasin and Katzir 2016.

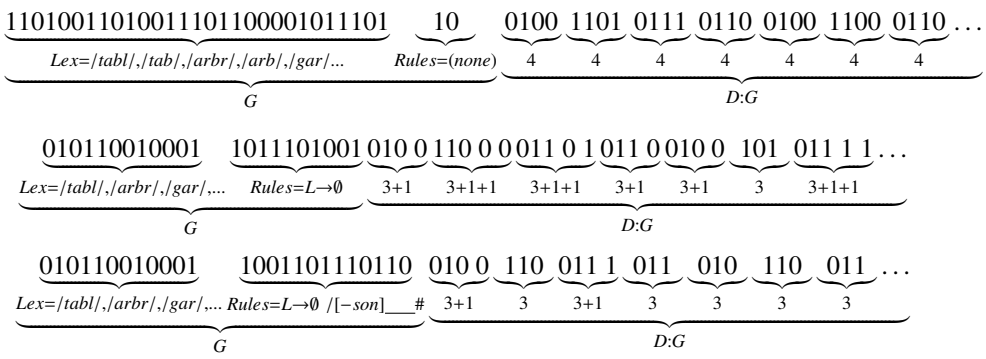


Figure 1: Schematic illustration of three hypotheses. (The order of URs in the lexicon and of tokens in $D : G$ are unrelated.) Introducing a naive lexicon (*top*), in which [tabl] and [tab] have distinct URs results in a complex grammar. Capturing optional *L*-deletion with (2) allows the grammar to be simplified (*middle*): the complexity of the rule is outweighed by the savings of eliminating unnecessary URs. Moreover, since there are now fewer URs than with the naive lexicon, each UR can be specified more succinctly. However, an additional bit is needed for specifying the actual surface form of each occurrence of *L* in a UR (for each surface token of that UR). Finally, restricting the context of *L*-deletion, using (3), allows us to limit the extra bit to just those URs that require it (*bottom*): /tabl/ but not /gar/.

is weighed against the added complexity in stating a grammar that does that while still accounting for both [tabl] and [tab]. In the present case, a grammar that rules out [sab] will be quite complex: it might dispense with *L*-deletion and resort to memorizing each observed surface form using a separate UR; or it might state a highly involved rule (or system of rules) that license *L*-deletion in those forms where both variants of a pair has been observed. Either way, the result will be a complex grammar that does not justify the minimal savings obtained by not having to specify whether *L*-deletion has applied for the single occasion when the UR /sabl/ was chosen. (This is very different from the case of [gar], where preventing inappropriate *L*-deletion involved only a slight increase in grammar size, and where there were sufficiently many relevant instances of *L* in non-deleting environments to justify the added complexity.) Consequently, the accidental gap arising from seeing an occurrence of [sabl] without an instance of [sab] will not prevent the MDL learner from keeping the rule of *L*-deletion in (5), thus generalizing beyond the data, which seems to be the correct result.

Suppose now that the learner sees not just one instance of [sabl] but rather many instances, still without any instance of [sab]. In this case, the absence of [sab] will start looking less like an accident of the specific data sequence seen so far and more like a systematic fact of French that needs to be captured. The MDL learner allows us to make this intuition precise: with sufficiently many occurrences of [sabl], the extra bit that is needed to state for each occurrence that /sabl/ does not undergo optional *L*-deletion results in an increase to $|D : G|$ that is big enough to justify blocking *L*-deletion for this UR. How exactly *L*-deletion is blocked will depend on the representations available to the learner. For example, if these representations offer a general way to mark exceptions to rules, the learner might choose to mark /sabl/ as an exception to

L-deletion. If such a method is not available, the learner might choose to block *L*-deletion in a more *ad hoc* way. For example, the learner might decide to add a special segment at the end of the UR (e.g., storing the relevant UR as /sablx/), thus preventing the *L* under consideration from appearing in the right context for deletion, along with a rule that deletes that special segment and is ordered after *L*-deletion.

Before proceeding, we note that in the discussion above we assumed that the input to the learner is a sequence of surface forms of words in isolation. If further information is available to the learner, such as the order of words in sentences or representations of scenes in which words are uttered, the decision of the learner regarding which forms to collapse using phonological rules can change. For example, a learner considering a small portion of the English lexicon containing 'spare', 'pear', 'spit', 'pit', 'stick', 'tick', and similar pairs might mistakenly collapse these pairs with the aid of an optional rule of [s]-deletion before [p] word-initially. By considering not just words in isolation but also the linguistic and extra-linguistic contexts in which they appear, however, an MDL learner will be justified in moving to a more complex grammar that does not collapse the relevant pairs but rather represents them using distinct URs in the lexicon.

The balancing of economy and restrictiveness has made MDL – and the closely related Bayesian approach to learning – helpful across a range of grammar induction tasks, in works such as Horning (1969), Berwick (1982), Ellison (1994), Rissanen and Ristad (1994), Stolcke (1994), Grünwald (1996), de Marcken (1996), Brent (1999), Clark (2001), Goldsmith (2001), and Dowman (2007), among others. Recently, Rasin and Katzir (2016) have used MDL to show how complete phonological grammars can be acquired distributionally within constraint-based phonology. The present work shows how the same can be done within rule-based phonology. In particular, we will show how the same MDL metric that supported the correct generalization in the case of the optional rule of *L*-deletion in French, as discussed above, will support the acquisition of whole phonological grammars, including the lexicon, the segmentation of forms into stems and affixes, a variety of phonological rules, and both transparent and opaque rule interactions. The simulations illustrating the use of MDL for the acquisition of phonological grammars – at present, using small corpora generated from artificial grammars – will be presented in section 3. Before that, in the remainder of the present section, we describe the phonological representations that we assume in order to make explicit their contribution to the MDL score, and we describe the search procedure we use to traverse the space of possible grammars.

2.2 Representations

As is standard, we assume that segments, both in phonological rules and in the lexicon, are represented not atomically but as feature bundles. For convenience, each simulation below works with a feature table that makes distinctions that are relevant to the phenomenon at hand, but we remain agnostic here as to whether learners start with a large innate table or acquire language-specific tables at an earlier stage. To illustrate, the feature table in Figure 2 will be used for those simulations that are based on English.

	cons	*voice*	*coronal*	*cont*	*low*	*high*	*back*	*son*	*lateral*	*labial*	*strident*
d	+	+	+	-	-	-	-	-	-	-	-
t	+	-	+	-	-	-	-	-	-	-	-
z	+	+	+	+	-	-	-	-	-	-	+
s	+	-	+	+	-	-	-	-	-	-	+
g	+	+	-	-	-	-	-	-	-	-	-
k	+	-	-	-	-	-	-	-	-	-	-
b	+	+	-	-	-	-	-	-	-	+	-
p	+	-	-	-	-	-	-	-	-	+	-
m	+	+	-	-	-	-	-	+	-	+	-
n	+	+	+	-	-	-	-	+	-	-	-
r	+	+	+	+	-	-	-	+	-	-	-
l	+	+	+	+	-	-	-	+	+	-	-
a	-	+	-	+	+	-	+	+	-	-	-
o	-	+	-	+	-	-	+	+	-	-	-
e	-	+	-	+	-	-	-	+	-	-	-
i	-	+	-	+	-	+	-	+	-	-	-
u	-	+	-	+	-	+	+	+	-	-	-

Figure 2: Feature table

2.2.1 Phonological rules

Feature bundles based on feature tables such as the one in Figure 2 are used to state the phonological rules. The general form of rules is as follows, where A, B are feature bundles or $\emptyset$; X, Y are (possibly empty) sequences of feature bundles; and *optional?* is a boolean variable specifying whether the rule is obligatory or optional (Figure 3).

$$\underbrace{A}_{\text{focus}} \rightarrow \underbrace{B}_{\text{change}} / \underbrace{X}_{\text{left context}} \_\_ \underbrace{Y}_{\text{right context}} \quad \text{(optional?)}$$

Figure 3: Rule format

The following, for example, is an optional phonological rule of vowel harmony that fronts a vowel before another front vowel when the two are separated by arbitrarily many consonants, stated in textbook notation in (6a) and in string notation (more convenient for the purposes of the conversion to bits below, and using various delimiters, marked with # with certain subscripts and discussed shortly) in (6b).[9]

(6) Vowel harmony rule

[9]String notation for phonological rules is familiar from early generative phonology (see especially Chomsky and Halle 1968 pp. 390ff.). As a reviewer notes, the implementation details of our notation are somewhat different from those in *SPE*. We will not investigate here the question of whether the precise choice of implementation in this case has testable empirical consequences.

a. Textbook notation

$$\left[-cons\right] \rightarrow \left[-back\right] / \_\_ \left[+cons\right]^{*} \begin{bmatrix} -cons \\ -back \end{bmatrix} \text{ (optional)}$$

b. String notation

$$-cons\#_{rc} - back\#_{rc}\#_{rc} + cons * \#_b - cons\#_f - back\#_{rc}1\#_{rc}$$

As discussed informally in section 2.1 above, determining both $|G|$ and $|D : G|$ for purposes of MDL is done in bits, where each bit represents a single binary choice. In the simple representations that we use in this paper, all possible outcomes at any particular choice point (whether binary or otherwise) are treated as equally easy to encode. For purposes of presentation, we will first discuss a particularly simple representation in which at any given choice point, the different outcomes are not just equally easy on average to encode but actually have fixed, equal length codes. This will allow us to discuss the various encodings in terms of fixed conversion tables in which if there are n possible outcomes, each will be assigned a code of length $\lceil \lg n \rceil$ bits. In our actual simulations, presented in section 3, we will deviate from the encoding presented below by allowing non-integral code lengths, taking $\lg n$ rather than $\lceil \lg n \rceil$ as the code length for an n-ary choice point.[10]

Within the simplified representational framework just described, determining the length in bits of a single phonological rule for the purposes of MDL is done by using a conversion table that states the codes for the possible elements within phonological rules. An example of a possible conversion table appears in Figure 4.

Symbol	Code	Symbol	Code
$\#_f$ (feature)	0000	cons	0110
$\#_b$ (bundle)	0001	voice	0111
$\#_{rc}$ (rule component)	0010	velar	1000
+	0011	back	1001
-	0100	...	...
*	0101	...	...

Figure 4: Conversion table for rules

Using the conversion table in Figure 4, we can encode the phonological rule of vowel harmony (in (6) above) by converting each element in the string representation in (6b) into bits according to Figure 4 and concatenating the codes. To ensure unique readability, we use delimiters to mark the end of the description of features within a feature bundle ($\#_f$), feature bundles within the left and right contexts of a rule ($\#_b$), and

[10]The reason for this change is that the encoding used in the current section, using $\lceil \lg n \rceil$, is highly sensitive to changes in which the number of outcomes at a given choice point crosses a power of 2 (which is where $\lceil \lg n \rceil$ changes). By taking $\lg n$ instead of $\lceil \lg n \rceil$, this unhelpful sensitivity to powers of 2 is avoided. On the other hand, using conversion tables with fixed code lengths, corresponding to $\lceil \lg n \rceil$, allows us to keep the presentation considerably simpler than if we had to discuss $\lg n$ in terms of code lengths. We therefore keep the presentationally simpler $\lceil \lg n \rceil$ for the current section and the more robust $\lg n$ for the actual simulations.

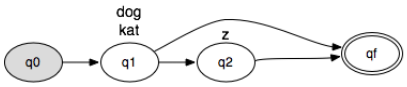


Figure 5: An HMM representation of a lexicon

the rule's components ($\#_{rc}$; in terms of the notation in figure 3, an occurrence of $\#_{rc}$ occurs after each of *A*, *B*, *X*, *Y*, and *optional?*). The following is the result, and its length is 73 bits:

(7) Vowel harmony rule (bit representation):

$$\underbrace{0100}_{-}\ \underbrace{0110}_{cons}\ \underbrace{0010}_{\#_{rc}}\ \underbrace{0100}_{-}\ \underbrace{1001}_{back}\ \underbrace{0010}_{\#_{rc}}\ \underbrace{0010}_{\#_{rc}}\ \underbrace{0011}_{+}\ \underbrace{0110}_{cons}\ \underbrace{0101}_{*}\ \underbrace{0001}_{\#_{b}}$$

$$\underbrace{0100}_{-}\ \underbrace{0110}_{cons}\ \underbrace{0000}_{\#_{f}}\ \underbrace{0100}_{-}\ \underbrace{1001}_{back}\ \underbrace{0010}_{\#_{rc}}\ \underbrace{1}_{1}\ \underbrace{0010}_{\#_{rc}}$$

A phonological rule system is a sequence of phonological rules. Since the encoding described above allows us to determine from the bit representation where each rule ends, we can specify a phonological rule system by concatenating the encodings of the individual rules while maintaining unique readability with no further delimiters. The ordering of the rules is the order in which they are specified, from left to right. At the end of the entire rule system another $\#_{rc}$ is added.

2.2.2 Lexicon

The lexicon contains the URs of all the possible morphemes. Since morphemes combine selectively and in specific orders, some information about morpheme combinations must be encoded. We encode this information using Hidden Markov Models (HMMs), where morphemes are listed in the emission table for specific states, and the possible combinations are defined by state transitions. A simple example is provided in Figure 5.

The HMM in Figure 5 defines a lexicon with two kinds of morphemes: the stems /dog/ and /kat/, and the optional suffix /z/. As with rules, description length is not calculated directly for the standard, graphical notation of the HMM but rather for a bit-string form. As before, we start with an intermediate string representation for the HMM, as presented in Figure 7 (derived from the concatenation of the string representations for the different states, as listed in Figure 6; the delimiter $\#_S$ marks the end of the list of outgoing edges from a state and $\#_w$ marks the end of each emitted word; another $\#_w$ is added at end of each state). Within the simplified representational framework described earlier, we convert the string to a bit-string using a conversion table, as in Figure 8. As before, all choices at a given point are uniform, with the same code length for all possible selections at that point ($\lceil \lg n \rceil$ if there are n possible choices). As discussed above, the actual simulations presented in section 3 use $\lg n$ rather than $\lceil \lg n \rceil$ as the code length.

state	encoding string
q_0	$q_0 q_1 \#_S \#_w$
q_1	$q_1 q_2 q_f \#_S \text{dog} \#_w \text{kat} \#_w \#_w$
q_2	$q_2 q_f \#_S \text{z} \#_w \#_w$

Figure 6: String representations of HMM states

$$q_0 q_1 \#_S \#_w \#_w q_1 q_2 q_f \#_S \text{dog} \#_w \text{kat} \#_w \#_w q_2 q_f \#_S \text{z} \#_w \#_w$$

Figure 7: String representation of an HMM

2.2.3 Data given the grammar

Turning to the encoding of the data given the grammar, *D*:*G*, recall that the generation of a surface form involves concatenating several morphemes in a specific order and applying a sequence of phonological rules. Given the grammar as described above, specifying a surface form will therefore involve: (a) specifying the sequence of morphemes (as a sequence of choices within the lexicon, repeatedly stating the code for a morpheme according to the table in the current state followed by the code to make the transition to the next state); and (b) specifying the code for each application of an optional rule. Note that obligatory rules do not require any statement to make them apply.

Our goal, given a surface form, is to determine the best way to derive it from the grammar in terms of code length. A naive approach to this parsing task would be to try all the ways to generate a surface form from the grammar. Even with simple grammars, however, this approach can be unfeasible. Instead, we compile the lexicon and the rules into a weighted finite-state transducer (FST) that allows us to obtain the best derivation using dynamic programming. The compilation of the rules relies on Kaplan and Kay (1994).

Let us illustrate the encoding of best derivations in the case of the form $[\text{k}^h\text{æts}]$ – actually, of the simpler $[\text{kæts}]$ – using the FSTs for two simple grammars. First, consider the FST in Figure 9, which corresponds to a grammar with the lexicon in Figure 10 and no phonological rules. Using this FST, encoding the word $[\text{k}^h\text{æts}]/[\text{kæts}]$ requires 16 bits. The initial transition from q_0 to q_1 is deterministic and costs zero bits. After that, each of the four segments costs four bits: three bits to specify the segment itself (since there are eight outgoing edges from q_1) followed by one bit to specify the transition from q_2 (loop back to q_1 or proceed to q_3). The encoding, using the conversion table in Figure 12, is in Figure 11.[11]

Consider now the more complex FST in Figure 13, which corresponds to a grammar with the lexicon in Figure 5 and the English voicing assimilation rule. This FST corresponds to a more restrictive grammar: differently from the simpler FST in Figure 9, the present FST can only generate a handful of surface forms. Consequently, the present FST offers a shorter *D*:*G*. Specifically, since specifying $[\text{k}^h\text{æts}]/[\text{kæts}]$ re-

[11] Specifying $[\text{k}^h\text{æts}]$ requires handling the aspiration of the initial segment. Since the relevant rule is obligatory, the same number of bits is required as for $[\text{kæts}]$, though the FST is slightly more complex.

State	Code
$\#_S$	000
q_0	001
q_1	010
q_2	011
q_f	100

Segment	Code
$\#_w$	0000
a	0001
k	0010
d	0011
...	...

Figure 8: Conversion table for HMM

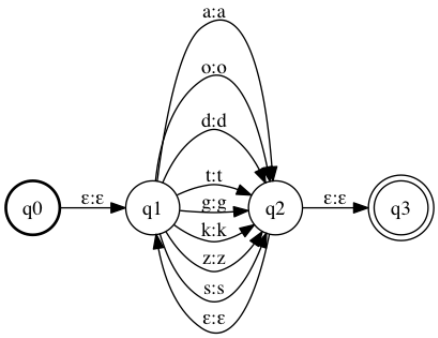


Figure 9: Naive FST

quires making only two choices in the FST, both of them binary, it allows us to encode the relevant string using only 2 bits, as in Figure 14.

2.3 Search

Above we saw how encoding length, $|G| + |D{:}G|$, is derived for any specific hypothesis G. In order to use it for learning, the learner can search through the space of possible hypotheses and look for a hypothesis that minimizes encoding length. We do not wish to make any claims about the search that the human learner might perform: our only claim in this paper concerns the MDL evaluation metric as a promising guide in comparing hypotheses. However, in order to show how this metric can guide the learner not just in the minimal comparisons discussed above but also when the learner faces a large space of possible hypotheses, we must combine the metric with some search procedure. Since the hypothesis space is big – infinitely so in principle – an exhaustive

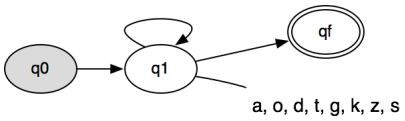


Figure 10: Lexicon corresponding to the naive FST

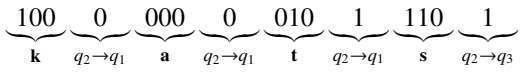


Figure 11: Encoding of a surface form using the naive FST

State q_0	
Arc	Code
(-,q_1)	ϵ

State q_1	
Arc	Code
(a,q_2)	000
(o,q_2)	001
(t,q_2)	010
(d,q_2)	011
...	...

State q_2	
Arc	Code
(-,q_1)	0
(-,q_3)	1

Figure 12: Conversion table for naive FST

search is out of the question, and a less naive option must be used. For concreteness, we adopt a genetic algorithm (GA), a general strategy that supports searching through complicated spaces that involve multiple local optima (Holland 1975).

The search starts with a random population of hypotheses that are generated by randomly selecting a lexicon and a set of ordered rules for each hypothesis. Individual hypotheses are selected for the next generation based on their fitness. The fitness of a hypothesis G equals $|G| + |D{:}G|$, the encoding length derived for it. Once a set of hypotheses is selected for the next generation, each pair of hypotheses is crossed-over to produce two offspring which replace their parents, and each offspring undergoes a random mutation to either its lexicon or its rule set. The simulation ends after a specified number of generations. The fittest hypothesis in the last generation is reported below as the final grammar.[12]

3 Simulations

The present section provides several simulations in which the MDL learner described in section 2 is faced with unanalyzed data exhibiting various linguistically-relevant patterns. We are not able to test the learner on real-life corpora at this point: both the size of the relevant part of the search space and the time it takes to parse each hypothesis during the search grow rapidly with the size and complexity of the corpus. Instead, we provide a proof-of-concept demonstration, using small datasets generated by artificial grammars that incorporate phonologically interesting dependencies. To simulate a larger corpus, we multiply $|D{:}G|$ by 10 in the simulations reported below (the effect is similar to presenting the learner with each word 10 times).[13] Section 3.1 illustrates our learner's acquisition of optionality, using a dataset based on the case of

[12]For a detailed discussion of the search procedure and the code for the simulations see Lan 2018.

[13]The one exception to the multiplication of $|D : G|$ by 10 is the simulations in section 3.1 for which we use different multipliers, as discussed below.

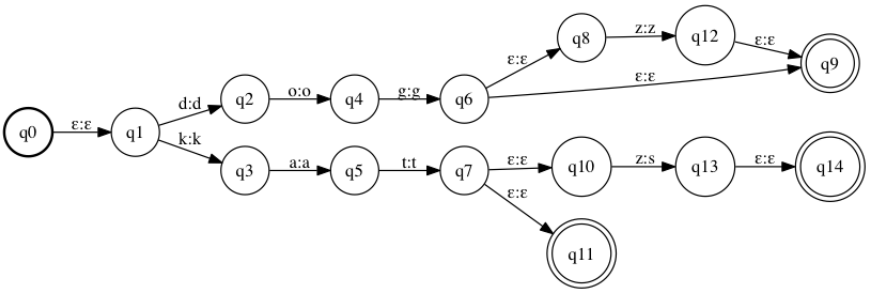


Figure 13: A more complex FST

$$\underbrace{0}_{q_1 \rightarrow q_2}\ \underbrace{0}_{q_6 \rightarrow q_8}$$

Figure 14: Encoding of a surface form using the more complex FST

optional French *L*-deletion discussed above. Section 3.2 uses a dataset based on /-z/-affixation in English to illustrate the joint acquisition of affixation and phonological processes. Section 3.3 extends the results of section 3.2 by showing how the learner can acquire two rules and their ordering in the case of transparent rule interaction. Section 3.3 modifies the English-based dataset to one that involves counterbleeding opacity and shows that the MDL learner succeeds in this case as well. Section 3.5 shows that the MDL learner succeeds on a case of counterfeeding opacity modeled after the interaction of two processes in Catalan.

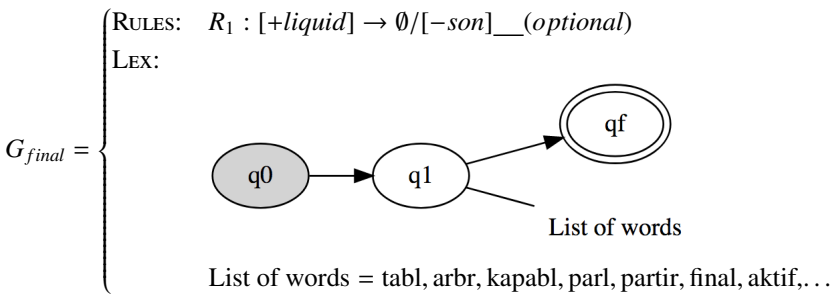
3.1 Optionality

The first dataset shows a pattern modeled after French *L*-deletion (Dell, 1981) and is designed to test the learner on the problem of restricted optionality. As discussed in section 2.1, the challenge for the learner is to strike the right balance between economy and restrictiveness. The learner needs to generalize beyond the data and conclude that for each pair like [tab]-[tabl] there is a single UR, and that a rule of *L*-deletion optionally applies. But the learner must not overgeneralize and should restrict *L*-deletion to only apply after obstruents, despite the added complexity of specifying the restricted environment in the description of the rule.

The data presented to the learner in the present simulation consisted of 91 words, including 33 collapsible pairs (since the task in our simulations is the acquisition of a grammar from distributional evidence alone, from the learner's perspective the data are an unstructured sequence of surface forms: the learner does not know that surface forms like [tab] and [tabl] are related in any way). A sample of the data is given in (8). Encoding length of the data given the grammar was multiplied by 50 and the encoding

length of the HMM was multiplied by 20.[14]

(8) tab, tabl, arb, arbr, kapab, kapabl, parl, partir, final, aktif, . . .



Description length: $|G_{final}| + |D{:}G_{final}|$ = 29,100.4 + 30,153.8 = 59,254.3

Figure 15: Final grammar for the French optionality simulation. The grammar includes the restricted *L*-deletion rule and forms like /tabl/ without their *L*-deleted counterparts (like /tab/). Here and below all scores are rounded to the first decimal place.

The learner induced the correct optional rule and converged on the target lexicon (Figure 15). Compared to the final (correct) grammar, the over-generating hypothesis has a shorter grammar but a longer *D*:*G*, leading to an overall longer description:

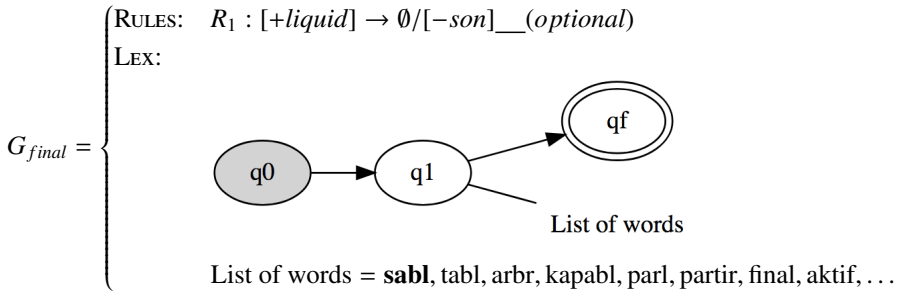
(9) a. Correct Hypothesis:
- R_1 : [+*liquid*] → ∅/[−*son*]__ (optional)
- Description length: |*G*| + |*D*:*G*| = 29,100.4 + 30,153.8 = 59,254.3

b. Over-generating Hypothesis:
- R_1 : [+*liquid*] → ∅/__ (optional)
- Description length: |*G*| + |*D*:*G*| = 29,092.9 + 33,141.5 = 62,234.5

In section 2.1 we discussed the undergeneralization problem for restrictiveness-only learning principles like the subset principle. We mentioned a scenario in which a learner has heard a surface form such as [sabl] but, accidentally, has not yet heard its *L*-elided variant [sab]. We noted that, while we would expect the human learner to generalize and learn *L*-deletion in the face of a single accidental gap, the subset principle predicts that *L*-deletion would be avoided. The MDL principle, on the other hand, predicts generalization. We ran another simulation of French using a variant of the corpus in (8) in which [sabl] was added without its *L*-elided variant [sab]. As expected, the learner generalized correctly and converged on the hypothesis in Figure

[14]The French simulation uses other parameters than all other simulations (where the encoding length of the data given the grammar was multiplied by 10 and the encoding length of the HMM was not multiplied by any factor). In the case of French, the search with the usual parameters did not converge. At present, we are not sure whether this is because the search was difficult in this case or because of something more significant.

16 which includes the *L*-deletion rule and a variant of the lexicon that also contains /sabl/.



Description length: $|G_{final}| + |D{:}G_{final}|$ = 29,517.5 + 30,610.1 = 60,127.6

Figure 16: Final grammar for a variant of the French-optionality simulation with an occurrence of [sabl] in the data but no occurrences of [sab]. The grammar includes the *L*-deletion rule which can generate the unattested [sab] as an output of /sabl/.

Consider now another situation discussed in section 2.1, where the learner encounters multiple occurrences of [sabl] – say, 100 occurrences – but still no occurrences of [sab]. In this situation, the absence of [sab] looks more systematic. The representations we have adopted for phonological rules do not allow for the direct statement of exceptions, but the learner can try to block *L*-deletion in various other ways. For example, the hypothesis in (10a) adds the dummy sonorant /m/ after /b/ to block *L*-deletion, only to later delete /m/ through the obligatory rule R_2. Despite this added complexity to the grammar, hypothesis (10a) has a shorter description length than the one in (10b), which does not mark /sabl/ as an exception and thus wastes bits on the encoding of the data (1 bit for each specification that optional *L*-deletion does not apply to /sabl/). Our actual simulation did not converge on (10a). Instead, it converged on a less optimal hypothesis that prevents *L*-deletion from applying to /tabl/ in other ways. Since the details are somewhat complicated, we refer the reader to the simulation logs available at https://github.com/taucompling/morphophonology_spe.

(10) a. Hypothesis with /sabl/ as an exception
- Lexicon: **sabml**, tabl, arbr, . . .
- Rules:
  - R_1 : [+*liquid*] → ∅/[−*son*]__ (optional)
  - R_2 : [+*cons*] → ∅/[−*son*]__[+*lateral*]
- Description length: |*G*| + |*D*:*G*| = 29,643.7 + 59,679.2 = 89,323.0

b. Hypothesis with no exception marks
- Lexicon: sabl, tabl, arbr, . . .
- R_1 : [+*liquid*] → ∅/[−*son*]__ (optional)
- Description length: |*G*| + |*D*:*G*| = 29,517.5 + 64,679.2 = 94,196.7

3.2 Joint learning of morphology and phonology

Our next simulation demonstrates the learner's ability to perform joint learning of morphology and a single phonological rule. Other works in the literature that perform joint learning of this kind include Naradowsky and Goldwater (2009) and (in a framework of constraint-based phonology) Rasin and Katzir (2016). After establishing this baseline, we will proceed, in the following sections, to the joint learning of morphology and rule interaction, a task that, as discussed in section 4, has not been accomplished in previous work. In the present simulation, the learner's tasks are to decompose the unanalyzed surface forms into a lexicon of underlying morphemes and to learn the relevant phonological rule.

Our example is modeled after English voicing assimilation where, as discussed in section 1, the suffix /z/ becomes voiceless following a voiceless consonant. The learner was presented with 250 words generated by creating all combinations of 25 verbal stems with 10 suffixes (including the null suffix) and applying voicing assimilation.[15] A sample of the data is provided in (11).

(11)

stem\suffix	∅	-z	-ing	-er	⋯
rent	rent	rents	renting	renter	
kontrol	kontrol	kontrolz	kontroling	kontroler	
glu	glu	gluz	gluing	gluer	
⋯					

The simulation converged on the grammar in Figure 17, which contains the correct rule and segmented lexicon. Given this grammar, generating a surface form requires first choosing a stem (out of 25 stems, at a cost of $\lg 25$ bits), then choosing a suffix (out of 10 suffixes, at a cost of $\lg 10$ bits), which makes a total of $\lg 25 + \lg 10 \approx 7.96$ bits for encoding each surface form. For comparison, consider the minimally-different alternative hypothesis in (12) that fails to learn the voicing-assimilation rule and stores both -z and -s as suffixes without collapsing them into a single UR. The hypothesis in (12) has a slightly smaller $|G|$: it stores an additional suffix in the lexicon (-s) but saves some space by omitting the rule. On the other hand, (12) over-generates. Any stem can be suffixed by either -z or -s regardless of the voicing of its final consonant. Thus, for example, both [rents] and [rentz] can be generated from the stem /rent/. This over-generation translates into a larger $|D{:}G|$: with the additional suffix, encoding any surface form given (12) now requires choosing a suffix out of 11 suffixes, so the total cost per surface form is $\lg 25 + \lg 11 \approx 8.1$ bits. Compared to the target hypothesis in Figure 17, the added cost of encoding each surface form given (12) is small (≈ 0.14 bits), but it accumulates over the entire corpus and ends up outweighing the slight advantage that (12) has in terms of $|G|$. Overall, then, the target hypothesis in Figure 17 wins due to a smaller combined $|G| + |D{:}G|$.

(12) Over-generating Hypothesis:

- Rules: ∅

[15] When attached to verbs, as in our simulation, the suffix /z/ marks the 3rd person singular in present tense. Since at present we do not model part-of-speech categories, our presentation of voicing assimilation will not distinguish this suffix from the nominal plural marker /z/.

- List of suffixes = z, s, . . .
- Description length: $|G|$ + $|D{:}G|$ = 804.4 + 20,258.2 = 21,062.6

In the simplified setting we have considered here, the corpus includes all combinations of 25 stems and 10 suffixes (a total of 250 words). This means, for example, that a hypothesis that simply memorizes the data (without performing any segmentation or learning any rules) would be as successful as the target hypothesis in terms of tightness of fit to the data, as both hypotheses generate precisely the same set of forms. In terms of $|D{:}G|$, encoding each surface form given the memorizing hypothesis would require choosing one out of 250 words in the lexicon at a cost of $\lg 250$ bits. Since $\lg 250 = \lg 25 + \lg 10$, this cost is identical to the cost given the target hypothesis. Despite the tie in the value for $|D{:}G|$, the target hypothesis wins due to its strictly smaller $|G|$. In a more realistic setting, the corpus will typically contain gaps, which would give the memorizing hypothesis an advantage in terms of $|D{:}G|$. For example, if five stem+suffix combinations (e.g., [kontrol-er]) are missing from the corpus, encoding a surface form given the memorizing hypothesis would cost $\lg 245$ bits, compared to an unchanged cost of $\lg 250$ for the target hypothesis (which can generate the five unattested combinations). As the data D grows, this wastefulness of the target hypothesis would accumulate and at some point outweigh the savings in the lexicon obtained by segmenting D. To estimate the effect of an increase in D, we created a variant of the data in (11) by omitting five words chosen at random, and we calculated different values for $|G|$ + $|D{:}G|$ while varying the multiplier for $|D{:}G|$. We found that when the multiplier for $|D{:}G|$ exceeds 1,039, the target hypothesis loses to the memorizing hypothesis in terms of the combined $|G|$ + $|D{:}G|$. We re-ran the simulation several times with the gapped corpus using each of the following multipliers for $|D{:}G|$: 10, 100, 1,000, 10,000, and 100,000. The simulation converged on the target hypothesis in Figure 17 in all cases. At least for the cases of the multipliers 10,000 and 100,000, this means that the simulation converged on a sub-optimal hypothesis. Since this is an accident of the search procedure, whose modeling is not our focus in this paper (as mentioned in section 2.3), we leave attempts to optimize the results with larger multipliers to a separate occasion.

3.3 Rule Ordering

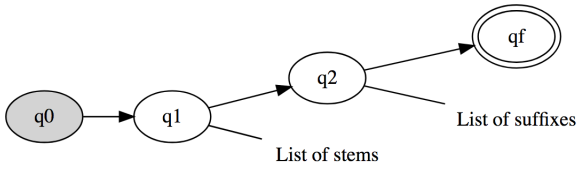
Rule-based phonology accounts for the interaction of phonological processes through rule ordering. In English, as we have seen, voicing assimilation devoices the suffix /-z/ when preceded by a voiceless obstruent. Epenthesis inserts the vowel [ɪ] between two sibilants (as in [glæsɪz] ,'glasses'). To derive forms such as [glæsɪz], where voicing assimilation does not apply and the suffix remains voiced, epenthesis is ordered before assimilation. When epenthesis applies to the UR /glæs-z/, it disrupts the adjacency between the suffix and the preceding consonant, rendering assimilation inapplicable. The opposite ordering would have derived the incorrect form *[glæsɪs], as demonstrated in (13):

(13) a. Good: epenthesis before assimilation

G_{final} =

RULES: $R_1 : [-son] \rightarrow [-voice] / [-voice]$ __

LEX:



List of stems = rent, klaimb, glu, komit, straik, drim, kontrol, . . .

List of suffixes = ϵ, z, ing, er, al, abl, ment, i, . . .

Description length: $|G_{final}| + |D{:}G_{final}|$ = 837.1 + 19,914.4 = 20,751.5

Figure 17: Final grammar for the joint learning simulation. The grammar includes the voicing assimilation rule and a segmented lexicon with the UR /-z/ from which both surface [-z] and [-s] can be derived.

	/glæs-z/
Epenthesis	glæsɪz
Assimilation	-
	[glæsɪz]

b. Bad: assimilation before epenthesis

	/glæs-z/
Assimilation	glæss
Epenthesis	glæsɪs
	*[glæsɪs]

Our next dataset was generated by an artificial grammar modeled after the interaction of voicing assimilation and epenthesis in English. The learner was presented with 250 words generated by creating the same combinations of stems and suffixes as in the previous section and applying epenthesis (14a) and voicing assimilation (14b), in this order. A sample of the data is provided in (15). The learner converged on the expected lexicon and on the two rules – epenthesis (R_1) and assimilation (R_2) – and their correct ordering (Figure 18).

(14) Rules

a. Rule 1: [i]-epenthesis between stridents

b. Rule 2: Progressive assimilation of [$-voice$] (to an adjacent segment)

(15)

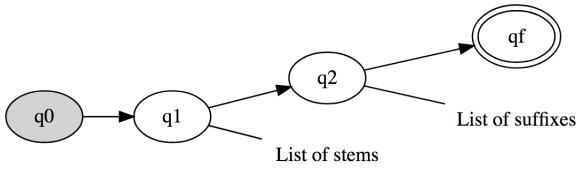
stem\suffix	∅	-z	-ing	-er	⋯
rent	rent	rents	renting	renter	
klaimb	klaimb	klaimbz	klaimbing	klaimber	
kros	kros	krosiz	krosing	kroser	
⋯					

$G_{final} =$

RULES: $R_1 : \emptyset \rightarrow [+high, -back] / [+strident]$ __ $[+strident]$

$R_2 : [-son] \rightarrow [-voice] / [-voice]$ __

LEX:



List of stems = rent, klaimb, glu, komit, straik, drim, kontrol, . . .

List of suffixes = ϵ, z, ing, er, al, abl, ment, i, . . .

Description length: $|G_{final}| + |D{:}G_{final}| = 894.1 + 19{,}914.4 = 20{,}808.5$

Figure 18: Final grammar for the rule-ordering simulation. The grammar includes epenthesis and voicing assimilation, in this order, and a segmented lexicon.

3.4 Counterbleeding opacity

The term *opacity* is used to describe rules whose effect is obscured on the surface, often because of an interaction with another rule (Kiparsky 1971, Baković 2011). One type of opacity called *counterbleeding* in the literature results when a rule R_2 removes the conditions for the application of another rule R_1 which has applied earlier in the derivation. R_1 is opaque since its environment of application is missing on the surface.

Our next dataset was designed to test the learner on the problem of counterbleeding opacity. We used two rules modeled after English epenthesis and voicing assimilation and changed the order such that assimilation was ordered first:

(16) Rules
   a. Rule 1: Progressive assimilation of [−*voice*] (to an adjacent segment)
   b. Rule 2: [i]-epenthesis between stridents

The result is that feature spreading takes place even between segments that are separated by an epenthetic vowel on the surface. Examples of natural languages that reportedly show a similar interaction between feature spreading and epenthesis are some varieties of English and Armenian, as reported in Vaux (2016), and Iraqi Arabic, as reported in Kiparsky (2000, citing Erwin, 1963).

As shown in (17), the opposite rule ordering would lead to the wrong result. Given the correct order, epenthesis applies after assimilation, rendering assimilation opaque: the first consonant of the suffix undergoes assimilation but is preceded by the epenthetic vowel on the surface.

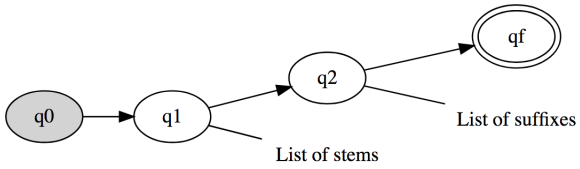
(17) Voicing assimilation crucially precedes epenthesis
   a. Good: assimilation before epenthesis

$G_{final} =$ {

RULES: $R_1 : [-son] \rightarrow [-voice] / [-voice]$ __

$R_2 : \emptyset \rightarrow [+high, -back] / [+strident]$ __ $[+strident]$

LEX:



List of stems = rent, klaimb, glu, komit, straik, drim, kontrol, . . .

List of suffixes = ϵ, z, ing, er, al, abl, ment, i, . . .

Description length: $|G_{final}| + |D{:}G_{final}| = 894.1 + 19{,}914.4 = 20{,}808.5$

Figure 19: Final grammar for the counterbleeding opacity simulation. The grammar includes voicing assimilation and epenthesis, in this order, and a segmented lexicon.

	/glæs-z/
Assimilation	glæss
Epenthesis	glæsɪs
	[glæsɪs]

b. Bad: epenthesis before assimilation

	/glæs-z/
Epenthesis	glæsɪz
Assimilation	-
	*[glæsɪz]

For this simulation, the dataset was generated by taking the same combinations of 25 stems and 10 suffixes as before and applying voicing assimilation and epenthesis, in this order. A sample of the data is provided in (18). The learner converged on the expected lexicon and on the two rules – assimilation (R_1) and epenthesis (R_2) – and their correct ordering (Figure 19).

(18)

stem\suffix	∅	-z	-ing	-er	⋯
rent	rent	rents	renting	renter	
klaimb	klaimb	klaimbz	klaimbing	klaimber	
kros	kros	krosis	krosing	kroser	
⋯					

3.5 Counterfeeding opacity

The type of opacity called *counterfeeding* in the literature results when a rule R_2 creates the conditions for the application of another rule R_1 which has applied earlier in the

derivation. R_1 is opaque since it does not apply even though its conditions of application are met on the surface. In Catalan (Mascaró 1976), for example (and simplifying), nasals are deleted word-finally (19a) and a rule of cluster simplification deletes a stop word-finally after a nasal (19b) and creates the environment for final-nasal deletion, which does not apply on the surface in (19b).

(19) a. kuzí ~ kuzín-s 'cousin.SG ~ cousin.PL'
b. kələ́n ~ kələ́nt-ə 'hot.MASC ~ hot.FEM'

Our next dataset was designed to test the learner on the problem of counterfeeding opacity. We used two rules modeled after final-nasal deletion and cluster simplification in Catalan. We generated 65 words by creating all combinations of 13 stems and 5 suffixes (all are actual Catalan morphemes) and applying final-nasal deletion and cluster simplification, in this order (20). Word boundaries were represented directly using the special symbol '#' which we added to the feature table along with the feature *boundary* ('#' had a positive value for *boundary* and a negative value for every other feature). A sample of the data is given in (21). The learner converged on a segmented lexicon and on the two rules – final-nasal deletion (R_1) and cluster simplification (R_2) – and their correct ordering, as in Figure 20.

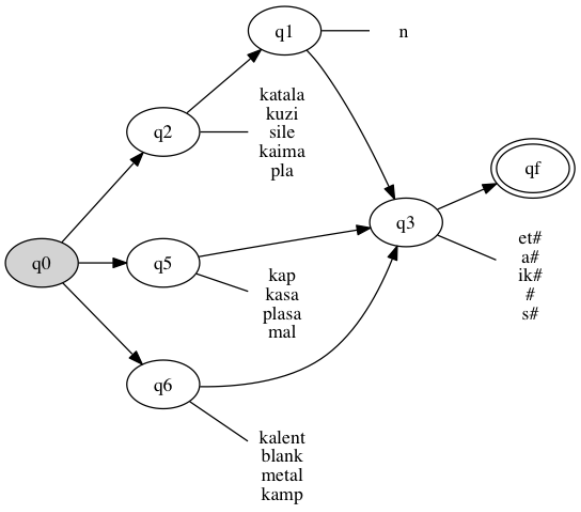
(20) Rules
a. Rule 1: Delete a nasal word-finally
b. Rule 2: Delete a word-final stop following a nasal

(21)

stem\suffix	∅	-s	-et	⋯
kalent	kalen#	kalents#	kalentet#	
kuzin	kuzi#	kuzins#	kuzinet#	
⋯				

While the learner has arrived at what seem like adequate morphological decomposition, phonological rules, and their ordering, a closer look at the final hypothesis in Figure 20 reveals three differences from the hypothesis we expected. First, the learned rule of cluster simplification deletes stops in a broader environment: before a word boundary or any other stop (not just before a word boundary). This rule is as simple as our target rule, which had [+boundary] as its right environment ([−*cont*] → ∅/[+*nasal*] __ [+*boundary*]). This is a result of our encoding scheme, in which the word boundary is part of the feature table and is specified as [−*cont*] (the single feature [+boundary] is as simple to state as [-cont]). Our corpus did not include any nasal-stop-stop sequences, which meant that the broader rule was consistent with the data. A second property of the final hypothesis we would like to mention is that the 13 stems are split between three states (q_2, q_5, q_6). As it turns out, this split is a sophisticated way for the learner to obtain minor savings in $|D{:}G|$. Without this split, encoding the choice of each of the stems (as part of the encoding of a surface form) would cost $\lg 13 \approx 3.7$ bits. Given the split, it becomes cheaper to encode the choice of each of the stems in states q_5 and q_6: $\lg 3$ (to encode the move from q_0) + $\lg 4$ (to encode the choice of one out of four stems in either q_5 or q_6), which equals $\lg 12 \approx 3.58$

$$G_{final} = \begin{cases} \textsc{Rules}: & R_1 : [+nasal] \rightarrow \emptyset / \ \_\_ \ [+boundary] \\ & R_2 : [-cont] \rightarrow \emptyset / [+nasal] \ \_\_ \ [-cont] \\ \textsc{Lex}: & \end{cases}$$



Description length: $|G_{final}| + |D{:}G_{final}| = 517.3 + 4{,}031.1 = 4{,}548.5$

Figure 20: Final grammar for the counterfeeding opacity simulation. The grammar includes final-nasal deletion and cluster simplification (in this order) and a segmented lexicon. Word boundaries are represented directly using the symbol '#'.

bits, a slightly lower cost than 3.7. These savings outweigh the added cost of encoding the choice of each of the stems in q_2, as well as the added cost of representing two additional states in the lexicon, which makes the split beneficial in terms of compression. The third property of the final hypothesis we would like to mention is the representation of /n/-final stems as a concatenation of their /n/-less variants (q_2) with /n/ (q_1). Since this hypothesis only adds /n/ to /n/-final stems, it generates exactly the same forms as our target hypothesis (which stores /n/-final stems with their final /n/), so there is no difference between the two in terms of |*D*:*G*|. The final grammar is slightly more complex than the target grammar (it costs ≈ 5 additional bits). In other words, the split of /n/-final stems was a harmless accident of the search procedure.

4 Previous work on learning rule-based phonology

We presented a learner that uses the MDL evaluation metric, which minimizes |*G*| + |*D*:*G*|, to jointly learn morphology and phonology within a rule-based framework. This learner is fully distributional, working from unanalyzed surface forms alone – without access to paradigms or negative evidence – to obtain the URs in the lexicon, the possible morphological combinations, and the ordered phonological rules. It acquires both allophonic rules and alternations, and for a rule of the form $A \rightarrow B / X\_\_Y$ it can arrive at generalizations both in terms of the focus and the change (*A* and *B*, respectively) and in terms of the context (*X* and *Y*). And it handles both optionality and rule interaction, including instances of opacity. In this section we review past work on inducing rule-based phonology and highlight aspects of the task handled by our learner that were left open in the literature.

As we discussed in section 2.1 above, evaluation metrics that do not balance |*G*| against |*D* : *G*| have not been successful. In particular, and as discussed by Dell (1981) and others, the evaluation metric of SPE, which aimed at minimizing |*G*|, leads to overgeneralization. We further showed how a restrictiveness metric, which can be stated in terms of minimizing |*D* : *G*|, addresses the problem for the SPE metric but does so at the cost of blocking valid generalization. Not surprisingly, neither of these two evaluation metrics have led to actual learners.

Johnson (1984) offers the first working learner for phonological rule systems. It is particularly significant since it can handle the task of learning rule interactions, including cases of opacity. Differently from Chomsky and Halle's approach and the present one, Johnson's learner is based not on an evaluation metric that compares hypotheses given the data but rather on a procedure that obtains contexts for individual phonological rules. In particular, when *A* and *B* alternate, Johnson's procedure examines the contexts in which *A* appears and those in which *B* appears; for the rule $A \rightarrow B / X\_\_Y$, a context $X\_\_Y$ is obtained (not necessarily uniquely) by considering what is common to all the contexts in which *B* appears and different from every context in which *A* appears. The alternating segments *A* and *B* themselves are identified with the help of morphologically analyzed paradigms, which the procedure assumes as input. The learner is thus not fully distributional. The dependence on morphological analysis to identify *A* and *B* also means that the procedure is aimed at alternations and cannot generally acquire cases of allophony that are not identifiable from alternations. It also

generalizes only in terms of the context $X\_\_Y$ and provides no handle on generalizations in terms of A or B. Finally, by relying on contexts in which B appears but A does not, the procedure misses cases of optionality, which by definition involve contexts where both A and B can appear.

Johnson (1984)'s learner can be seen as the direct predecessor of the procedure-based learner for rule-based phonology proposed by Albright and Hayes (2002, 2003). Like Johnson's learner, Albright and Hayes's learner assumes that morphological paradigms are identified in advance and is thus not fully distributional.[16] For Albright and Hayes, paradigms serve a similar role in morphology to the role they served for Johnson in phonology, namely the identification of change in an alternation, leaving the learner the task of finding the context for the change. Albright and Hayes then add a step of phonological acquisition in which the learner examines the morphological changes obtained so far and checks whether a given morphological change can apply even when superficially inappropriate by adding a phonological rule. During phonological induction, the set of possible contexts for phonological rules is provided in advance (rather than acquired) in the form of phonotactically illicit sequences. Like Johnson (1984), Albright and Hayes's learner is aimed at alternations and cannot generally acquire cases of allophony that are not identifiable from alternations. Moreover, it does not provide a handle on generalizations in terms of A and B or on optionality, and it does not acquire rule interactions.

A different procedure-based learner was proposed by Gildea and Jurafsky (1995, 1996), who adapt Oncina et al. (1993)'s OSTIA model for the induction of certain deterministic finite-state transducers (FSTs) – specifically, subsequential FSTs – to the task of acquiring phonology.[17] OSTIA starts from an FST that faithfully maps inputs to outputs and gradually merges states in the FST while maintaining subsequentiality, and Gildea and Jurafsky enhance this process with linguistically-motivated constraints to obtain linguistically-natural mappings of URs to surface forms. Since the procedure requires the URs to be given in advance, however, it is not distributional. Like Johnson (1984), it also generalizes entirely in terms of the context $X\_\_Y$ not in terms of A or B. It also has no handle on optionality (though Gildea and Jurafsky suggest that a stochastic HMM merger framework, for example along the lines of Stolcke and Omohundro 1993, might address this).[18]

Of the learners for rule-based phonology in the literature, our learner is closest

[16] See Dunbar (2008) and Simpson (2010) for later procedure-based learners for aspects of morphophonology. Like Albright and Hayes's learner, these proposals rely on pre-analyzed paradigmatic pairs as input to the learner and are thus not distributional.

[17] Thus, while aiming at phonological rule systems, Gildea and Jurafsky (1995, 1996) do not learn such systems directly but rather FSTs, which are a rather different kind of representation. In fact, FSTs are a computationally convenient form into which one can compile both rule-based phonology (see Kaplan and Kay 1994) and constraint-based phonology (see Frank and Satta 1998 and Riggle 2004). See Cotterell et al. (2015) for a recent learner for FSTs that, while not siding with either rule-based or constraint-based phonology is closer in spirit to the latter. We should note that Gildea and Jurafsky's goal is not the modeling of the acquisition of rule-based phonology as such but rather an investigation of the role of linguistic biases in this kind of learning. In particular, they show that three quite general biases improve the acquisition of rule-based grammars within Oncina et al.'s framework.

[18] It is difficult to evaluate the suitability of the model to rule interaction. Gildea and Jurafsky (1995, 1996) provide an example with multiple rules, but these rules do not interact, and it remains unclear whether rule interaction (and, in particular, opacity) can be handled by their system.

to those proposed by Goldwater and Johnson (2004), Goldsmith (2006), and Naradowsky and Goldwater (2009). All three are fully distributional learners for rule-based morpho-phonology that, like Chomsky and Halle (1968), rely on an evaluation metric rather than on a procedural approach.[19] Differently from Chomsky and Halle (1968) – and similarly to the present proposal – these learners use a balanced evaluation metric that optimizes economy and restrictiveness simultaneously.[20] Goldwater and Johnson (2004)'s algorithm starts with a morphological analysis based on Goldsmith (2001)'s MDL-based learner and then searches for phonological rules that lead to an improved grammar, where the improvement criterion is Bayesian. Goldsmith (2006)'s learner follows a similar path but uses MDL also for the task of phonological learning. Naradowsky and Goldwater (2009)'s learner is a variant of Goldwater and Johnson (2004)'s learner with joint learning of morphology and phonology, thus addressing (similarly to the present learner) the interdependency of phonology and morphology. As stated, all three learners can acquire rules only at morpheme boundaries, which, as in the learners of Johnson (1984), Albright and Hayes (2002, 2003), and Simpson (2010), limits considerably the phonological rules that they learn. Like these procedural learners, the three balanced learners generalize only with respect to $X\_\_Y$ and not with respect to A and B. They are also aimed at obligatory rules and do not handle rule interaction. One way of interpreting our simulations above is as showing that these limitations are not essential within this framework and that a balanced evaluation metric can support the acquisition of allophony, generalizations over both the context and the change, optionality, and rule interactions.

A final comparison for the current proposal is with the procedural learner of Calamaro and Jarosz (2015), which learns phonological rules – both allophony and alternations – in a fully distributional way by extending the allophonic learner of Peperkamp et al. (2006). Peperkamp et al. detect maximally dissimilar contexts as hints for allophonic distribution. For example, [æ] and [æ̃] are allophones in English, and the contexts that they can appear in are very different: [æ̃] can only appear before a nasal consonant, while [æ] can only appear elsewhere. Peperkamp et al. provide a statistical score that identifies such dissimilarities in the contexts in which two segments can appear; when two segments have highly dissimilar contexts, they are considered to be potential allophones.[21] Calamaro and Jarosz (2015) look to extend Peperkamp et al. (2006)'s model beyond allophony, in order to account for neutralization processes. The challenge, given Peperkamp et al.'s dissimilarity score, is that neutralization involves segments whose possible contexts may have a significant overlap. Consider, for example, a language like Dutch that has final devoicing. In such a language, [t] and [d]

[19] Naradowsky and Goldwater (2009) targets orthographic rules rather than phonology, but the difference is immaterial.

[20] Outside of rule-based phonology, Cotterell et al. (2015) and Rasin and Katzir (2016) propose balanced learners for the acquisition of phonology, the former within a phonological framework of weighted edits and the latter within constraint-based phonology.

[21] This raises all the usual issues with phonemics, such as the fact that, in English, [h] and [ŋ] are in complementary distribution but are not phonemically related. And indeed, Peperkamp et al. encounter many false positives (even more so since they do not require full complementary distribution). Echoing early structuralist proposals, they propose that complementarity should be combined with requirements of phonological similarity. As discussed by Chomsky (1964, p. 85), such requirements do not resolve the problem for phonemic analysis.

might contrast everywhere except for the context __#; a global score of contextual dissimilarity will consequently treat [t] and [d] as quite similar and fail to relate them to one another. In order to overcome this challenge, Calamaro and Jarosz consider contextualized distributional dissimilarity: for a given context *X__Y* and two potential alternants *A* and *B*, they compute a dissimilarity score for the triple < *X__Y*, *A*, *B* > by comparing the probability of the context *X__Y* given *A* and given *B*. These dissimilarity scores are summed for the context and for the featural change over all pairs *A* and *B* that have that change, thus allowing for generalization in terms of the change. A further extension introduces generalization over contexts (subject to two special conditions). In terms of comparison with the present proposal, Calamaro and Jarosz's model faces two challenges that, as far as we can tell, are hard to address within the framework of distribution comparison that they adopt. First, their model does not handle rule orderings. This gap is particularly difficult to bridge in the case of opaque rule interactions, where surface distributions obscure the correct context for rule application. The second challenge to Calamaro and Jarosz model concerns optionality. When a rule is optional, the distribution of *A* and *B* can be similar in all contexts, so a dissimilarity detector will fail to identify the rule.

5 Discussion

We presented an MDL-based learner for the unsupervised joint learning of lexicon, morphological segmentation, and ordered phonological rules from unanalyzed surface forms. The learner contributes to the literature on learning rule-based morpho-phonology, a literature that starts with Chomsky and Halle (1968) and continues with Johnson (1984), Albright and Hayes (2002, 2003), Gildea and Jurafsky (1995, 1996), Goldwater and Johnson (2004), Naradowsky and Goldwater (2009), and Calamaro and Jarosz (2015), among others. The current learner goes beyond the literature in two main respects. First, it can handle rule systems that involve not just obligatory rules but also optional ones.[22] And second, it can handle rule interaction, including cases of opacity. In handling both optionality and rule interaction the present proposal offers what to our knowledge is the first distributional learner that can acquire a full morpho-phonological rule system with the structure proposed in the phonological literature. However, the present work has focused on small, artificial corpora that exhibit specific morpho-phonological patterns, and it remains to be seen if and how the approach can extend to larger, more realistic corpora.

The proposed learner uses the simple and very general MDL approach, in which hypotheses are compared in terms of two readily available quantities: the storage space required for the current grammar and the storage space required for the current grammar's best parse of the grammar. It has been argued recently that this approach has cognitive plausibility as a null hypothesis for language learning in humans and that it offers a reasonable framework for the comparison of different representational choices

[22] In the literature on constraint-based phonology, which we do not discuss in the present paper, the acquisition of optionality has received a fair amount of attention. See Coetzee and Pater 2011 for review and discussion. We note that proposals within this literature, such as Boersma and Hayes 2001, generally assume that the learner is given information about URs and is therefore not fully distributional.

in terms of predictions about learning (Katzir, 2014). From an empirical perspective, Pycha et al. (2003) have provided evidence that simplicity plays a central role in the acquisition of phonological rules.[23] If correct, the present work is a step toward a cognitively plausible learner for rule-based morpho-phonology, and its predictions can be compared with those of MDL or Bayesian learners for other representation choices such as Rasin and Katzir (2016)'s MDL learner for constraint-based phonology. We leave the investigation of such predictions for future work.

References

Albright, Adam, and Bruce Hayes. 2002. Modeling english past tense intuitions with minimal generalization. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, 58–69. Association for Computational Linguistics.

Albright, Adam, and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: a computational/experimental study. *Cognition* 90:119–161.

Baker, Carl L. 1979. Syntactic theory and the projection problem. *Linguistic Inquiry* 10:533–581.

Baković, Eric. 2011. Opacity and ordering. In *The handbook of phonological theory, second edition*, 40–67. Wiley-Blackwell.

Berwick, Robert C. 1982. Locality principles and the acquisition of syntactic knowledge. Doctoral Dissertation, MIT, Cambridge, MA.

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

Boersma, Paul, and Bruce Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32:45–86.

Braine, Martin D. S. 1971. On two types of models of the internalization of grammars. In *The ontogenesis of grammar*, ed. D. J. Slobin, 153–186. Academic Press.

Brent, Michael. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Computational Linguistics* 34:71–105.

Calamaro, Shira, and Gaja Jarosz. 2015. Learning general phonological rules from distributional information: A computational model. *Cognitive Science* 39:647–666.

Chaitin, Gregory J. 1966. On the length of programs for computing finite binary sequences. *Journal of the ACM* 13:547–569.

Chomsky, Noam. 1964. *Current issues in linguistic theory*. Mouton & Company.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English*. New York: Harper and Row Publishers.

Clark, Alexander. 2001. Unsupervised language acquisition: Theory and practice. Doctoral Dissertation, University of Sussex.

Coetzee, Andries, and Joe Pater. 2011. The place of variation in phonological theory. In

[23] See also Moreton and Pater (2012a,b) for simplicity in phonological learning (though see Moreton et al. 2017 for an argument that phonotactic and concept learning are guided by something closer to a Maximum Entropy model rather than by simplicity), and see Goodman et al. (2008) and Orbán et al. (2008), among others, for empirical evidence for balanced learning elsewhere in cognition.

The handbook of phonological theory, ed. John Goldsmith, Jason Riggle, and Alan C. L. Yu, chapter 13, 401–434. Wiley-Blackwell.

Cotterell, Ryan, Nanyun Peng, and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics* 3:433–447.

Dell, François. 1981. On the learnability of optional phonological rules. *Linguistic Inquiry* 12:31–37.

Dowman, Mike. 2007. Minimum description length as a solution to the problem of generalization in syntactic theory. Ms., University of Tokyo, Under review.

Dunbar, Ewan. 2008. The acquisition of morphophonology under a derivational theory: A basic framework and simulation results. Master's thesis, University of Toronto.

Ellison, Timothy Mark. 1994. The machine learning of phonological structure. Doctoral Dissertation, University of Western Australia.

Endress, Ansgar D., and Marc D. Hauser. 2011. The influence of type and token frequency on the acquisition of affixation patterns: Implications for language processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 37:77–95.

Erwin, Wallace M. 1963. *A short reference grammar of Iraqi Arabic*. Georgetown University Press.

Frank, Robert, and Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24:307–315.

Gildea, Daniel, and Daniel Jurafsky. 1995. Automatic induction of finite state transducers for simple phonological rules. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 9–15. Association for Computational Linguistics.

Gildea, Daniel, and Daniel Jurafsky. 1996. Learning bias and phonological-rule induction. *Computational Linguistics* 22:497–530.

Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics* 27:153–198.

Goldsmith, John. 2006. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering* 12:1–19.

Goldwater, S., T. Griffiths, and M. Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. *Advances in neural information processing systems* 18:459.

Goldwater, Sharon, and Mark Johnson. 2004. Priors in Bayesian learning of phonological rules. In *7th Annual Meeting of the ACL Special Interest Group on Computational Phonology*, 35–42.

Goodman, N.D., J.B. Tenenbaum, J. Feldman, and T.L. Griffiths. 2008. A rational analysis of rule-based concept learning. *Cognitive Science* 32:108–154.

Grünwald, Peter. 1996. A minimum description length approach to grammar inference. In *Connectionist, statistical and symbolic approaches to learning for natural language processing*, ed. G. S. S. Wermter and E. Riloff, Springer Lecture Notes in Artificial Intelligence, 203–216. Springer.

Holland, John H. 1975. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.*. U Michigan Press.

Hopcroft, John E, Rajeev Motwani, and Jeffrey D Ullman. 2007. *Introduction to automata theory, languages, and computation author: John e. hopcroft, rajeev motwani, jeffrey*. Addison Wesley, 3rd edition.

Horning, James. 1969. A study of grammatical inference. Doctoral Dissertation, Stanford.

Johnson, Douglas. 1972. *Formal aspects of phonological description*. The Hague: Mouton.

Johnson, Mark. 1984. A discovery procedure for certain phonological rules. In *Proceedings of 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, 344–347.

Kaplan, Ronald M., and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.

Katzir, Roni. 2014. A cognitively plausible model for grammar induction. *Journal of Language Modelling* 2:213–248.

Kiparsky, Paul. 1971. Historical linguistics. In *A survey of linguistic science*, ed. W. O. Dingwall, 576–642. University of Maryland Linguistics Program, College Park.

Kiparsky, Paul. 2000. Opacity and cyclicity. *The Linguistic Review* 17:351–366.

Kolmogorov, Andrei Nikolaevic. 1965. Three approaches to the quantitative definition of information. *Problems of Information Transmission (Problemy Peredachi Informatsii)* 1:1–7. Republished as Kolmogorov (1968).

Kolmogorov, Andrei Nikolaevic. 1968. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics* 2:157–168.

Lan, Nur. 2018. Learning morpho-phonology using the Minimum Description Length principle and a genetic algorithm. Master's thesis, Tel Aviv University.

de Marcken, Carl. 1996. Unsupervised language acquisition. Doctoral Dissertation, MIT, Cambridge, MA.

Mascaró, Joan. 1976. Catalan phonology and the phonological cycle. Doctoral Dissertation, MIT.

Moreton, Elliott, and Joe Pater. 2012a. Structure and substance in artificial-phonology learning, part i: Structure. *Language and Linguistics Compass* 6:686–701.

Moreton, Elliott, and Joe Pater. 2012b. Structure and substance in artificial-phonology learning, part ii: Substance. *Language and Linguistics Compass* 6:702–718.

Moreton, Elliott, Joe Pater, and Katya Pertsova. 2017. Phonological concept learning. *Cognitive Science* 41:4–69.

Naradowsky, Jason, and Sharon Goldwater. 2009. Improving morphology induction by learning spelling rules. In *IJCAI*, 1531–1536.

Oncina, J., P. García, and E. Vidal. 1993. Learning subsequential transducers for pattern recognition interpretation tasks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 15:448–458.

Orbán, Gergő, József Fiser, Richard N Aslin, and Máté Lengyel. 2008. Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences* 105:2745–2750.

Peperkamp, Sharon, Rozenn Le Calvez, Jean-Pierre Nadal, and Emmanuel Dupoux. 2006. The acquisition of allophonic rules: Statistical learning with linguistic constraints. *Cognition* 101:B31–B41.

Prince, Alan, and Paul Smolensky. 1993. Optimality theory: Constraint interaction

in generative grammar. Technical report, Rutgers University, Center for Cognitive Science.

Pycha, Anne, Pawel Nowak, Eurie Shin, and Ryan Shosted. 2003. Phonological rule-learning and its implications for a theory of vowel harmony. In *Proceedings of the 22nd West Coast Conference on Formal Linguistics*, volume 22, 101–114. Somerville, MA: Cascadilla Press.

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

Riggle, Jason. 2004. Generation, recognition, and learning in finite state Optimality Theory. Doctoral Dissertation, UCLA, Los Angeles, CA.

Rissanen, Jorma. 1978. Modeling by shortest data description. *Automatica* 14:465–471.

Rissanen, Jorma, and Eric Sven Ristad. 1994. Language acquisition in the MDL framework. In *Language computations: DIMACS Workshop on Human Language, March 20-22, 1992*, 149. Amer Mathematical Society.

Simpson, Marc. 2010. From alternations to ordered rules: A system for learning derivational phonology. Master's thesis, Concordia University, Montreal.

Smolensky, Paul. 1996. The initial state and 'richness of the base' in Optimality Theory. Technical Report JHU-CogSci-96-4, Johns Hopkins University.

Solomonoff, Ray J. 1964. A formal theory of inductive inference, parts I and II. *Information and Control* 7:1–22, 224–254.

Stolcke, Andreas. 1994. Bayesian learning of probabilistic language models. Doctoral Dissertation, University of California at Berkeley, Berkeley, California.

Stolcke, Andreas, and Stephen Omohundro. 1993. Hidden Markov Model induction by Bayesian model merging. In *Advances in neural information processing systems*.

Vaux, Bert. 2016. Can epenthesis counterbleed assimilation? Talk presented at NAPhC 9, Concordia University, May 7-8, 2016.

Wallace, Christopher S., and David M. Boulton. 1968. An information measure for classification. *Computer Journal* 11:185–194.

Wexler, Kenneth, and Rita M. Manzini. 1987. Parameters and learnability in binding theory. In *Parameter setting*, ed. Thomas Roeper and Edwin Williams, 41–76. Dordrecht, The Netherlands: D. Reidel Publishing Company.

Yang, Charles. 2016. *The price of linguistic productivity: How children learn to break the rules of language*. MIT Press.