# Learning rule-based morpho-phonology

Ezer Rasin, Iddo Berger, and Roni Katzir

December 20, 2015

## 1 Introduction

Speakers' knowledge of the sound pattern of their language – their knowledge of morpho-phonology – goes well beyond the plain phonetic forms of words. In 'cats', pronounced [kʰæts], speakers of English know that the aspiration of the initial [k] and the voicelessness of the final [s] are no accident: voiceless stops such as [k] are always aspirated in the beginning of a syllable in English, and the expression of the plural morpheme is always the voiceless [s] after a voiceless stop such as [t]. Thus, imaginable forms such as [kæts] or [kʰætz] are not possible in English. In 'dogs', pronounced [dɔgz], it is similarly no accident that the initial d is unaspirated and that the final consonant is voiced: [dʰɔgz] and [dɔgs] are not possible. Since speakers have judgments about aspiration and voicing in novel forms, the relevant knowledge is systematic and cannot be accounted for by the simple memorization of surface forms. It is also specific to English, so it must be learned.

According to a long-standing model in linguistics, morpho-phonological knowledge is distributed between a lexicon with morphemes, usually referred to as Underlying Representations (URs), and context-sensitive rewrite rules that transform URs to surface forms (Chomsky and Halle, 1968).[1] For the example above, the URs would be /kæt/, /dɔg/, and /z/.[2] The phonological rules would be: (a) a rule that aspirates unvoiced stops in the beginning of a syllable; and (b) a rule that assimilates the voicing of a morpheme-initial /z/ to that of the final segment of the preceding morpheme. These two rules are both obligatory, but there are also optional rules. For example, the French word *table* 'table' can surface as either [tabl] or [tab]; the relevant phonological rule, which we return to below, must therefore be optional. As noted by Dell (1981) and others, optionality raises a variety of learning challenges. A further common learning challenge that we will examine below involves interacting rules, with one rule applying first and either creating or destroying the conditions for the application of a subsequent rule.

[1] Following Prince and Smolensky (1993)'s introduction of Optimality Theory (OT), much recent work has taken the phonological component to consist of surface-oriented constraints rather than rewrite rules. OT has given rise to a large literature on learning. We will not attempt to review the arguments that aim at choosing between rule-based and constraint-based phonology (see Vaux (2008), among others, for discussion). The learning approach presented below is also compatible with OT. See Rasin and Katzir (2015).

[2] Following phonological convention, we enclose surface forms with brackets and URs with slashes, so, for example, we will write that an underlying /kæt-z/ surfaces as [kʰæts].

As highlighted recently by Calamaro and Jarosz (2015), the child needs to acquire both the URs and the phonological rules in an unsupervised fashion, using distributional cues alone; moreover, phonological and morphological learning are interdependent (and may be acquired at around the same developmental stage), so learners cannot assume that morphological analysis has been completed by the time they acquire phonological rules. Chomsky and Halle (1968) already provide a proposal for learning, using an objective function that favors short grammars over longer ones. Several works over the years – see Johnson (1984), Gildea and Jurafsky (1995, 1996), Goldwater and Johnson (2004), Naradowsky and Goldwater (2009), Simpson (2010), Calamaro and Jarosz (2015), and Cotterell et al. (2015) – have presented learning algorithms that handle parts of the learning task. To date, however, significant aspects of the task remain an open challenge. In particular, no distributional learner in the literature can acquire optional rules or rules that interact.

In this paper we provide what to our knowledge is the first distributional learner that acquires both URs and phonological rules, including both optionality and rule interaction. Our learner is based on the principle of Minimum Description Length (MDL; Rissanen 1978; see also Wallace and Boulton 1968) which – like the closely related Bayesian approach – aims at balancing the complexity of the grammar and its fit of the data. We present the details of our learner in section 2, focusing on the representations used to state phonological rules. In section 3 we present learning simulations with optionality, rule interaction, and interdependent phonology and morphology. Section 4 discusses previous works on the learning of rule-based morpho-phonology. Section 5 concludes.

# 2 The present work

## 2.1 The MDL criterion

The principle of MDL aims at minimizing the overall description of the data, taking into account both the complexity of the grammar with that of the grammar's account of the data. The roots of MDL are in the pioneering work of Solomonoff (1964). MDL and closely related Bayesian approaches have been used for grammar induction in the works of 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), and Rasin and Katzir (2015), among others. The learner attempts to minimize the overall description of the data, measured in bits. The overall description is broken down into $G$, the encoding of the grammar (which, for SPE, includes both the lexicon and the rules), and $D$:$G$, the description of the data $D$ given the grammar. The length of $G$, $|G|$, corresponds to the informal notion of *economy*, familiar from the evaluation metric of Chomsky and Halle (1968): a grammar that requires fewer bits to encode is generally a simpler, less stipulative grammar. Meanwhile, the length of $D$:$G$, $|D$:$G|$, corresponds to *restrictiveness*: a grammar that requires fewer bits to encode the data is a grammar that considers the data typical and deviations from the data surprising. Minimizing the sum of both lengths balances economy and restrictiveness:

(1) $\arg\min_G\{|G| + |D{:}G|\}$

The combination of grammar and data is schematized in Figure 1 (modified from Rasin and Katzir 2015).

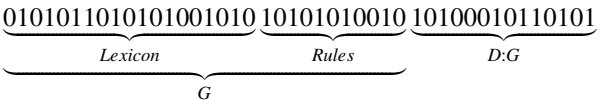



Figure 1: Schematic view of a phonological grammar and the data it encodes. The grammar $G$ consists of both lexicon and rules. The data $D$ are represented not directly but as encoded by $G$. The overall description of the data is the combination of $G$ and $D{:}G$.

## 2.2 Representations

As is standard in phonology, we assume that segments, both in phonological rules and in the lexicon, are represented not atomically but as feature bundles. Specifically, we will assume the following feature table, though we remain agnostic here as to whether this table is innate or acquired in an earlier stage (Figure 2)

| | *cons* | *voice* | *velar* | *cont* | *back* |
|---|---|---|---|---|---|
| d | + | + | - | - | - |
| t | + | - | - | - | - |
| g | + | + | + | - | - |
| k | + | - | + | - | - |
| z | + | + | - | + | - |
| s | + | - | - | + | - |
| i | - | + | - | + | - |
| u | - | + | - | + | + |

Figure 2: Feature table

### 2.2.1 Phonological rules

Feature bundles based on the table above 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)

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 a consonant, stated in textbook notation in (2a) and in string notation (more convenient for the purposes of the conversion to bits below) in (2b).

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

Figure 3: Rule format

(2) Vowel harmony rule

a. Textbook notation

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

b. String notation

$$-cons\#_{rc} - back\#_{rc}\#_{rc} + cons\#_{b} - cons\#_{f} - back\#_{rc}1\#_{rc}$$

Determining the length of the rule for the purposes of MDL is done 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. The representation scheme we use here treats all possible outcomes at any particular choice point as equally easy to encode. For the conversion table, this means that if there are $n$ possible elements that can appear within a rule, each will be assigned a code of length $\lceil \lg n \rceil$ bits.

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

Figure 4: Conversion table for rules

Using the conversion table in Figure 4, we can now encode the phonological rule of vowel harmony (in (2) above) by converting each element in the string representation in (2b) into bits according to Figure 4 and concatenating the codes. The following is the result, and its length is 69 bits:

(3) Vowel harmony rule (bit representation):

$$\underbrace{0100}_{-}\ \underbrace{0101}_{cons}\ \underbrace{0010}_{\#_{rc}}\ \underbrace{0100}_{-}\ \underbrace{1000}_{back}\ \underbrace{0010}_{\#_{rc}}\ \underbrace{0010}_{\#_{rc}}\ \underbrace{0011}_{+}\ \underbrace{0101}_{cons}\ \underbrace{0001}_{\#_{b}}$$

$$\underbrace{0100}_{-}\ \underbrace{0101}_{cons}\ \underbrace{0000}_{\#_{f}}\ \underbrace{0100}_{-}\ \underbrace{1000}_{back}\ \underbrace{0010}_{\#_{rc}}\ \underbrace{1}_{1}\ \underbrace{0010}_{\#_{rc}}$$

A phonological rule system is a sequence of phonological rules. Since each rule ends with the code for optionality followed by $\#_{rc}$, we can specify a phonological rule system by concatenating the encodings of the individual rules while maintaining unique

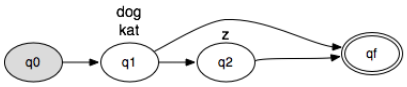



Figure 5: An HMM representation of a lexicon

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). We then 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.

| state | encoding string |
|---|---|
| $q_0$ | $q_0 q_1 \#_S \#_w$ |
| $q_1$ | $q_1 q_2 q_f \#_S$ dog$\#_w$kat$\#_w \#_w$ |
| $q_2$ | $q_2 q_f \#_S$ 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,

| 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

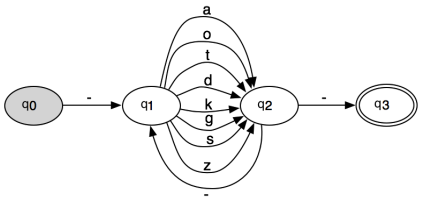



Figure 9: Naive FSA

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 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 finite-state automaton (FSA) that allows us to obtain the best derivation using dynamic programming. The compilation of the rules relies on Kaplan and Kay (1994); see Berger (2015) for detailed discussion.

Let us illustrate the encoding of best derivations in the case of the form $[\mathrm{k}^h\mathrm{æts}]$ – actually, of the simpler $[\mathrm{kæts}]$ – using the FSAs for two simple grammars. First, consider the FSA in Figure 9, which corresponds to a grammar with the lexicon in Figure 10 and no phonological rules. Using this FSA, encoding the word $[\mathrm{k}^h\mathrm{æts}]/[\mathrm{kæts}]$ 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 $q_2$. The encoding, using the conversion table in Figure 12, is in Figure 11.[3]

Consider now the more complex FSA in Figure 13, which corresponds to a grammar with the lexicon in Figure 5 and the English voicing assimilation rule. This FSA

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

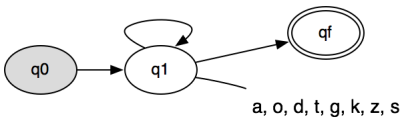



Figure 10: Lexicon corresponding to the naive FSA

$$\underbrace{100}_{\mathbf{k}}\ \underbrace{0}_{q_2 \to q_1}\ \underbrace{000}_{\mathbf{a}}\ \underbrace{0}_{q_2 \to q_1}\ \underbrace{010}_{\mathbf{t}}\ \underbrace{1}_{q_2 \to q_1}\ \underbrace{110}_{\mathbf{s}}\ \underbrace{1}_{q_2 \to q_3}$$

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

| State 0 | |
|---|---|
| Arc | Code |
| (-,1) | $\epsilon$ |

| State 1 | |
|---|---|
| Arc | Code |
| (a,2) | 000 |
| (o,2) | 001 |
| (t,2) | 010 |
| (d,2) | 011 |
| ... | ... |

| State 2 | |
|---|---|
| Arc | Code |
| (-,1) | 0 |
| (-,3) | 1 |

Figure 12: Conversion table for naive FSA

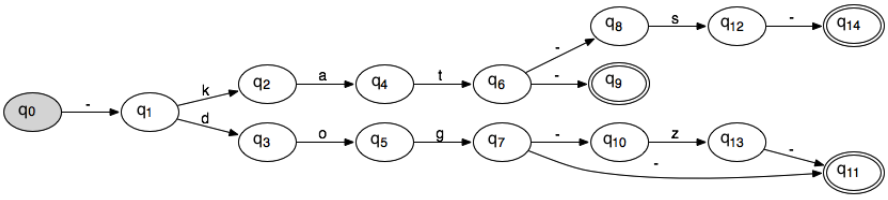



Figure 13: A more complex FSA


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

corresponds to a more restrictive grammar: differently from the simpler FSA in Figure 9, the present FSA can only generate a handful of surface forms. Consequently, the present FSA offers a shorter *D*:*G*. Specifically, since specifying $[\mathrm{k}^h\mathrm{æts}]/[\mathrm{kæts}]$ requires making only two choices in the FSA, 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. Since the hypothesis space is big – infinitely so in principle – an exhaustive search is out of the question, and a less naive option must be used. We adopt Simulated Annealing (SA; Kirkpatrick et al., 1983), a general strategy that supports searching through complicated spaces that involve multiple local optima.

SA proceeds by comparing a current hypothesis to its neighbors in terms of their goodness, which in our case is the total description length. That is if a current hypothesis $G$ has $G'$ as its neighbor, $|G| + |D{:}G|$ is compared to $|G'| + |D{:}G'|$. If $G'$ is better than $G$, the search switches to $G'$. Otherwise, the choice of whether to switch to $G'$ is made probabilistically and depends both on how much worse $G'$ is and on a *temperature* parameter. The higher the temperature, the more likely the search is to switch to a bad neighbor. The temperature is initially set to a relatively high value, and it is gradually lowered (by multiplying the temperature at each step by a constant $\alpha$ to yield the temperature at the next step) as the search progresses, making the search increasingly greedy. The search ends when the temperature descends below a fixed threshold. The specific parameters of the search are provided below.

For any grammar $G$, the neighbor grammar $G'$ is generated as a variant of $G$ in which one of the following changes occurs.

(4) Possible mutations during the search:
   a. Mutate rule: add/remove/modify rule, add/remove/modify feature bundle
   b. Mutate lexicon: add/remove/modify morpheme, add/remove/modify state

The possible mutations are described in greater detail in Appendix A.

The search starts from a naive initial hypothesis – generally far away from the optimal one – that can generate any sequence of segments. This hypothesis has the trivial lexicon in Figure 15 and no phonological rules.

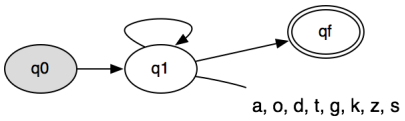



Figure 15: Initial hypothesis

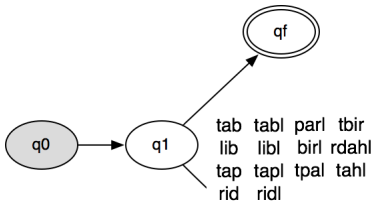
# 3 Simulations

## 3.1 Optionality

The first dataset shows a pattern modeled after French l-deletion and is designed to test the learner on the problem of restricted optionality. As noted by Dell (1981), the environment for optional l-deletion is restricted: the rule may apply, word-finally, following obstruents such as [b] (thus [tabl]~[tab]) but never after sonorants such as [r] (thus [parl] but never *[par]). This pattern is compatible with two different generalizations: application of l-deletion only after obstruents or application after any consonant, including sonorants. French speakers reach the first, restricted generalization over the second over-generating one, despite the absence of negative evidence suggesting non-application following sonorants (e.g., *[par]). Moreover, the restricted generalization corresponds to a more complex rule compared to the over-generating one, since specifying the environment as a consonant which is also a sonorant requires more features as part of the rule description than merely specifying a consonant ($\begin{bmatrix} +cons \\ -son \end{bmatrix}$ vs. $\left[+cons\right]$) or no context at all. French speakers reach the more complex rule.[4] The tradeoff between grammar economy and restrictiveness makes the French pattern a natural test case for a compression-based metric: minimizing the size of the grammar, |*G*|, will push the learner towards economizing the lexicon and collapsing pairs of lexical items (e.g., /tabl/, /tab/) into one (/tabl/) and deriving the other member through l-deletion. Minimizing the size of the grammar would generally be beneficial unless it is counterbalanced by an increased length of data encoding given the grammar, |*D*:*G*|. Having to make more choices in the face of optionality results in such an increase: encoding [tabl] as the output of /tabl/ requires one bit of information to specify its choice over [tab] whenever /tabl/ is selected from the lexicon. Collapsing [tabl] and [tab] into a single UR thus requires one bit to be specified whenever the relevant UR is selected, thus increasing |*D*:*G*|. However, collapsing the two forms into a single UR also means that the lexicon is now slightly smaller, which lowers |*G*| and – more importantly – can make the specification of the new UR slightly shorter than specifying either of the two

[4] In Dell's original paper, the rules are stated using non-linear representations and the difference in complexity comes from the assumption that referring to obstruents requires reference to syllable structure, while referring to consonants does not. Here we use linear representations, where the added complexity translates into an additional feature required to specify obstruents over consonants. But linear representations also make it possible to refer to obstruents using a single feature $\left[-son\right]$. To keep the structure of the problem faithful to Dell's version, the feature table we use for this simulation includes a segment [h] specified as a non-consonantal obstruent which does not trigger l-deletion. This makes a hypothetical rule that refers to obstruents using a single feature $\left[-son\right]$ over-generate, as it would incorrectly apply to [h].

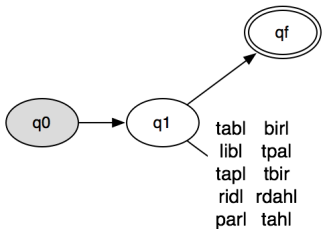
$G_{initial} =$ RULES: $\emptyset$; LEX:




Description length: $|G_{initial}| + |D{:}G_{initial}| = 5,980 + 2,100 = 8,080$

Figure 16: Initial grammar

$G_{final} =$ RULES: $R_1 : [+liquid] \rightarrow \emptyset / \begin{bmatrix} +cons \\ -son \end{bmatrix} \_\_\_ (optional)$; LEX:




Description length: $|G_{1,703,435}| + |D{:}G_{1,703,435}| = 4,745 + 2,200 = 6,945$

Figure 17: Final grammar

original ones, thus lowering $|D{:}G|$. On the other hand, the slight compression gained by eliminating one feature from the grammar ($\left[+son\right]$) would not justify paying additional bits (required to specify the chosen output) for every final l that follows a sonorant (as in /parl/, where compression in the lexicon is unavailable).

The data presented to the learner in the present simulation consisted of 14 words, including 4 collapsable pairs (5). The lexicon of the initial state was identical to the data and the rule set was empty (Figure 16). The parameters used in this simulation are: *initial_temperature*: 50, *cooling_rate*: 0.9999991, *threshold*: 1.0. Encoding length of the data given the grammar was multiplied by 50 and the encoding length of the HMM was multiplied by 20.

(5) tab, tabl, lib, libl, tap, tapl, rid, ridl, parl, birl, tpal, tbir, rdahl, tahl

The learner induced the correct optional rule and converged on the target lexicon (Figure 17). 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:

(6) a. Correct Hypothesis:

- $R_1 : [+liquid] \rightarrow \emptyset / \begin{bmatrix} +cons \\ -son \end{bmatrix}$ ___ (optional)
- Description length: $|G| + |D{:}G| = 4,745 + 2,200 = 6,945$

b. Over-generating Hypothesis:

- $R_1 : [+liquid] \rightarrow \emptyset / \begin{bmatrix} +cons \end{bmatrix}$ ___ (optional)
- Description length: $|G| + |D{:}G| = 4,733 + 2,400 = 7,133$

## 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 are Naradowsky and Goldwater (2009), Calamaro and Jarosz (2015), and (in a framework of constraint-based phonology) Rasin and Katzir (2015). 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 rule.

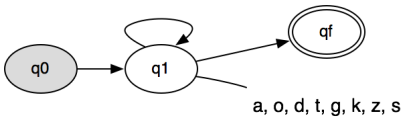
Our example is modeled after English voicing assimilation (where, as discussed in section 1, the plural suffix /z/ devoices following a voiceless obstruent). The learner was presented with 32 words generated by creating all combinations of 8 stems with 4 suffixes (including the null suffix) and applying voicing assimilation. A sample of the data is provided in (7). The parameters used in this simulation are: *initial_temperature*: 50, *cooling_rate*: 0.999995, *threshold*: 0.01. Encoding length of the data given the grammar was multiplied by 25.

(7)

| stem\suffix | ∅ | -gos | ⋯ |
|---|---|---|---|
| toz | toz | tozgos | |
| dot | dot | dotkos | |
| ⋯ | | | |

For the initial state of this simulation, we used a grammar that contained no rules and a lexicon that could generate any sequence of segments (Figure 18). The initial grammar can generate any possible surface string, and it treats all segments as equally likely. Since there is a total of 8 possible segments in this setting, specifying the choice of a single segment costs 3 bits of information, which makes a total of 3*l* bits for a surface form of length *l*. In the final grammar (Figure 19), surface forms were decomposed into stems and suffixes: generating a surface form requires first choosing one stem (out of 8 stems, at a cost of 3 bits), then choosing one suffix (out of 4 suffixes, including the null suffix, at a cost of 2 bits), which makes a total of only 5 bits per surface form.

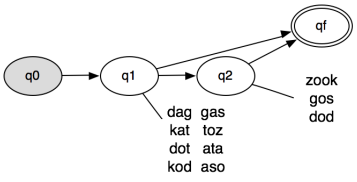
$$G_{initial} = \begin{cases} \text{RULES:} & \emptyset \\ \text{LEX:} & \end{cases}$$




Description length: $|G_{initial}| + |D{:}G_{initial}| = 97 + 17,600 = 17,697$

Figure 18: Initial grammar

$$G_{final} = \begin{cases} \text{RULES:} & R_1 : [+cons] \rightarrow [-voice] / [-voice]\_\_ \\ \text{LEX:} & \end{cases}$$




Description length: $|G_{final}| + |D{:}G_{final}| = 277 + 4,400 = 4,677$

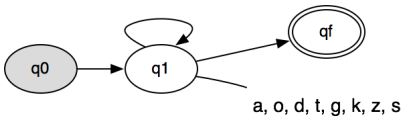
Figure 19: Final grammar

$|D{:}G|$ decreases dramatically from 17,697 in the initial state to 4,677 in the final state. The final lexicon stores stems and suffixes and is therefore more complex than the simple initial lexicon, but this addition is easily offset by the savings to $|D{:}G|$. Finally, the assimilation rule adds complexity to the set of rules, but it allows collapsing pairs of morphemes (like [-gos] and [-kos]) that differ minimally on the surface into a single underlying morpheme; compared to the rule-less alternative, this move decreases both $|G|$ – since it allows storing fewer items in the lexicon – and $|D{:}G|$ – since having fewer morphemes means that there are fewer choices to make in specifying a surface form.

## 3.3 Rule Ordering

Rule-based phonology accounts for the interaction of phonological processes through rule ordering. In English, voicing assimilation devoices the plural morpheme /-z/ when preceded by a voiceless obstruent (as in [kʰæts] ,'cats', but not in [dɔgz], 'dogs'). 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 plural morpheme remains voiced, epenthesis can be ordered before assimilation. When epenthesis applies to the UR /glæs-z/, it disrupts the adjacency between the plural morpheme and the preceding consonant, rendering assimilation inapplicable. The op-

$G_{initial} =$ {RULES: ∅; LEX:




Description length: $|G_{initial}| + |D{:}G_{initial}| = 97 + 29,800 = 29,897$

Figure 20: Initial grammar

posite ordering would have derived the incorrect form [glæsɪs], as demonstrated in (8):

(8) a. Good: epenthesis before assimilation

| | /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 50 words generated by creating all combinations of 10 stems with 5 suffixes (including the null suffix) and applying epenthesis (9a) and voicing assimilation (9b), in this order. A sample of the data is provided in (10). The parameters used in this simulation are: *initial_temperature*: 50, *cooling_rate*: 0.9999995, *threshold*: 1.0. Encoding length of the data given the grammar was multiplied by 25. The initial state was as before (Figure 20). The learner converged on the expected lexicon and rules (Figure 21).

(9) Rules

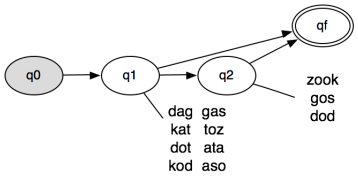
a. Rule 1: Low-vowel epenthesis between stridents

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

(10)

| stem\suffix | -gos | -zook | -sad | ⋯ |
|---|---|---|---|---|
| dag | daggos | dagzook | dagsad | |
| dot | dotkos | dotsook | dotsad | |
| gas | gaskos | gasazook | gasasad | |
| ⋯ | | | | |

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




Description length: $|G_{3,960,600}| + |D{:}G_{3,960,600}| = 402 + 6,300 = 6,702$

Figure 21: Final grammar

## 3.4 Opacity

The term *opacity* informally refers to a state of affairs where the effect of a rule is obscured on the surface, often because of an interaction with another rule. One type of opacity called *counterbleeding* in the literature results when a rule $R_2$ removes the environment of another rule $R_1$ which applies 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:

(11) Rules
   a. Rule 1: Progressive assimilation of [−*voice*] (to an adjacent segment)
   b. Rule 2: Low-vowel epenthesis between coronals

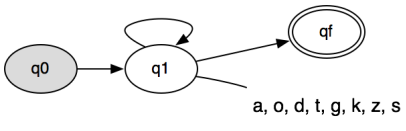
The result is that feature spreading takes place even between segments that are separated by an epenthetic vowel on the surface. Iraqi Arabic is an example of a natural language that shows a similar interaction between feature spreading and epenthesis, as reported in Kiparsky (2000, citing Erwin, 1963).

As shown in (12), 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.

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

| | /kat-zoka/ |
|---|---|
| Assimilation | katsoka |
| Epenthesis | katasoka |
| | [katasoka] |

$G_{initial} =$ {RULES: ∅; LEX:




Description length: $|G_{initial}| + |D{:}G_{initial}| = 97 + 35,300 = 35,397$

Figure 22: Initial grammar

b. Bad: epenthesis before assimilation

| | /kat-zoka/ |
|---|---|
| Epenthesis | katazoka |
| Assimilation | - |
| | *[katazoka] |

For this simulation, the learner was presented with 48 words generated by creating all combinations of 11 stems with 4 suffixes (including the null suffix) and applying voicing assimilation and epenthesis, in this order. A sample of the data is provided in (13). The parameters used in this simulation are: *initial_temperature*: 50, *cooling_rate*: 0.9999999, *threshold*: 1.0. Encoding of the data given the grammar was multiplied by 25. Given an initial state as before (Figure 22), the learner converged on the expected lexicon and rules (Figure 23)

(13)

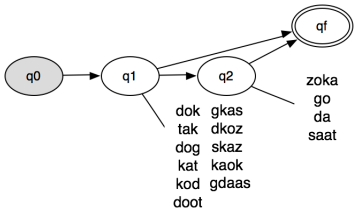
| stem\suffix | -zoka | -saat | ⋯ |
|---|---|---|---|
| dok | doksoka | doksaat | |
| kat | katasoka | katasaat | |
| ⋯ | | | |

# 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 morpho-phonology within a rule-based framework. This learner is fully distributional, working from unanalyzed surface forms 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 change ($A$ and $B$) and in terms of the context ($X$ and $Y$). And it handles both optionality and rule interaction, including counterbleeding 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.

Chomsky and Halle (1968) suggested that rule-based phonology can be learned using an evaluation metric that favors short grammars over complex ones: for a grammar

$$G_{final} = \begin{cases} \textsc{Rules}: & R_1 : [+cons] \rightarrow [-voice] / [-voice]\_\_ \\ & R_2 : \emptyset \rightarrow [+low] / \left[+coronal\right] \_\_ \left[+coronal\right] \\ \textsc{Lex}: & \end{cases}$$




Description length: $|G_{26,554,400}| + |D{:}G_{26,554,400}| = 402 + 7,175 = 7,585$

Figure 23: Final grammar

$G$ that can parse the data, the value of $G$ is the inverse of the length of $G$, $\frac{1}{|G|}$. Using this criterion, the child can try to search through the space of possible grammars, eliminating suboptimal grammars as it proceeds. That objective function was never used in an actual learning algorithm, however, and it was pointed out by Dell (1981) that it would yield incorrect results in various cases, including the case of optional l-deletion in French mentioned above. Specifically, recall from section 3.1 that the environment for l-deletion is restricted: the rule may apply following [b] (thus [tabl]~[tab]) but never after [r] (thus [parl] but never *[par]). Chomsky and Halle (1968)'s evaluation metric would have no reason to abandon a simple, overgenerating grammar that licenses *l*-deletion after any consonant in favor of a more complex one that restricts *l*-deletion to positions that follow obstruents.[5] Dell (1981) proposes to restrict the representations in phonology. As discussed in Rasin and Katzir (2015), however, the overgeneration problem for Chomsky and Halle (1968) is quite general: favoring simplicity alone allows simple, overgeneralizing grammars to win over complex, restrictive ones; what is needed is the balancing of economy with restrictiveness, as is done in MDL and Bayesian approaches of the kind presented in this paper.

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 $A$ appear 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 morpho-

[5] Similar challenges to Chomsky and Halle's evaluation metric outside of rule-based phonology were noted by Braine (1971) and Baker (1979).

logically 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 is the direct predecessor of the procedure-based learners for rule-based phonology proposed by Albright and Hayes (2002, 2003) and Simpson (2010). Like Johnson (1984), these learners assume that morphological paradigms are identified in advance and are thus not fully distributional. 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 not only the change – *A* and *B* – is given, but the set of possible contexts is provided as well in the form of phonotactically illicit sequences. Like Johnson (1984), Albright and Hayes (2002, 2003)'s learner is aimed at alternations and cannot generally acquire cases of allophony that are not identifiable from alternations, and the same is true for Simpson (2010)'s procedural learner as well. Similarly, neither Albright and Hayes (2002, 2003) nor Simpson (2010) provides a handle on generalizations in terms of *A* and *B* or on optionality.[6]

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 FSTs – specifically, subsequential FSTs – to the task of acquiring phonology.[7] 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

[6]As stated, Albright and Hayes (2002, 2003) and Simpson (2010) also do no handle rule interaction. However, it is conceivable that a variant of Johnson (1984)'s proposal for rule interaction could be adopted by these learners.

[7]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 to investigate 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.

merger framework, for example along the lines of Stolcke and Omohundro 1993, might address this).[8]

Of the learners for rule-based phonology in the literature, our learner is closest 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.[9] 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.[10] 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), prevents them from acquiring allophony. Like these procedural learners, the three balanced learners generalize only with respect to the context and not with respect to the change. 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 recent 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.[11] Since alternations do not involve complementary distribution, Calamaro and Jarosz (2015) 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

[8] 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 we cannot determine whether rule interaction (and, in particular, opacity) can be handled by their system.

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

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

[11] 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.

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). The model does not handle rule orderings, and we do not see how it might be extended to do so. It also fails on optionality, since when a rule is optional, the distribution of *A* and *B* can be similar in all contexts.

# 5 Discussion

We presented an MDL-based learner for the unsupervised joint learning of phonological rule systems and lexicons. 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), Gildea and Jurafsky (1995, 1996), Goldwater and Johnson (2004), Naradowsky and Goldwater (2009), Simpson (2010), Calamaro and Jarosz (2015), and Cotterell et al. (2015). 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. And second, it can handle rule interaction, including cases of opacity in which a rule destroys the environment of a rule that applies earlier. In handling both optionality and rule interaction the present proposal offers what to our knowledge is the first 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 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.[12] 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 (2015)'s MDL learner for constraint-based phonology. We leave the investigation of such predictions for future work.

[12] See also Moreton and Pater (2012a,b) for simplicity in phonological learning, and see Goodman et al. (2008) and Orbán et al. (2008) for empirical evidence for balanced learning elsewhere in cognition.

# Appendix

# A Mutations

This appendix describes all possible mutations used to generate a variant $G'$ of a hypothesis $G$ as part of the Simulated Annealing search procedure.

## A.1 Mutations on HMM

1. Combine emissions: Pick two emissions at random, concatenate them, and add the result to a random state
2. Clone emission: Pick an emission at random and add to a random inner state
3. Advance emission: Pick a random state $q_1$. From $q_1$, pick an emission and an outgoing state $q_2$ at random. Create a new state: $q'$. Add the chosen emission to $q'$. Remove the chosen emission from $q_1$. Add the transitions: $q_1$ to $q'$, $q'$ to $q_2$, and $q'$ to $q'$.
4. Add state: Add an empty state to the HMM (with no emissions or transitions)
5. Remove state: Remove a random state and all arcs connected to it
6. Add transition: Add a new transition between two random states (chosen with repetitions)
7. Remove transition: Remove a random transition from a random state
8. Add segment to emission: Add a random segment from the segment table to a random emission in a random position
9. Remove segment from emission: Remove a random segment from a random emission
10. Change segment in emission: Replace a random segment from a random emission with a different random segment
11. Add emission to state: Add a random segment from the segment table as a new emission to a random state
12. Remove emission from state: Remove a random emission from a random state

## A.2 Mutations on feature bundle list

1. Add feature bundle: Create a random feature bundle and insert it in a random position in the list
2. Remove feature bundle: Remove a feature bundle at a random position
3. Change existing feature bundle: Create a random feature bundle and mutate it using one of the mutations on feature bundles:

(a) Add feature: Add a random feature with a random value to the feature bundle

(b) Remove feature: Remove a feature at random from the feature bundle

(c) Change feature value: Flip the value of a random feature

### A.3 Mutations on rule set

1. Add rule: Generate a random rule with random feature bundles in each of the 4 parts of the rule: *change*, *target*, *left context*, and *right context*. Add the rule to the rule set

2. Remove rule: Remove a random rule from the rule set

3. Demote rule: Pick a random rule. Move it down in the rule order

4. Change rule:

   (a) Mutate target: mutate the *target* feature bundle

   (b) Mutate change: mutate the *change* feature bundle

   (c) Mutate left context: mutate the *left context* feature bundle list

   (d) Mutate right context: mutate the *right context* feature bundle list

   (e) Mutate obligatoriness: Flip the value of the obligatory value (which determines whether a rule is optional or obligatory)

## 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.

Berger, Iddo. 2015. Unsupervised induction of rule-based morpho-phonology. MA Thesis in preparation, Tel Aviv University.

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

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.

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.

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.

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

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, 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.

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

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. On the roles of markedness and contradiction in the use of alternatives. In *Semantics, pragmatics, and the case of scalar implicatures*, ed. Salvatore Pistoia Reda, 40–71. Palgrave-Macmillan.

Kiparsky, Paul. 2000. Opacity and cyclicity. *The Linguistic Review* 17:351–366.
Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220:671–680.
de Marcken, Carl. 1996. Unsupervised language acquisition. Doctoral Dissertation, MIT, Cambridge, MA.
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.
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. 2015. On evaluation metrics in Optimality Theory. To appear in *Linguistic Inquiry*, Vol. 47 Number 2.
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.
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. 2008. Why the phonological component must be serial and rule-based. In *Rules, constraints, and phonological phenomena*, ed. Andrew Nevins and Bert

Vaux, 20–60. Oxford: Oxford University Press.
Wallace, Christopher S., and David M. Boulton. 1968. An information measure for classification. *Computer Journal* 11:185–194.