Learning phonological optionality and opacity from distributional evidence*

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

Massachusetts Institute of Technology and Tel Aviv University

1. Introduction

Optionality and opacity pose obvious challenges for the child learning the morpho-phonology of their ambient language. In both cases, the already non-trivial task of identifying the phonological processes active in the language and their scope of application is made harder by the fact that these processes are not immediately recognizable from the surface forms observable by the child. Consider, for example, the case of optional word-final liquid deletion in French, discussed by Dell (1981): a word-final liquid may optionally delete (e.g., tabl ∼ tab 'table'), though only after an obstruent (thus, e.g., parl ∼ *par 'speak'). The relevant process can be stated as follows:[1]

(1) $L \rightarrow \emptyset$ /[−*son*]__# (optional)

Assuming that the child relies on *distributional* evidence alone – that is, unanalyzed surface forms, with no further assistance from paradigms or URs, let alone corrections from a tutor[2] – what will alert the child to the fact that liquids may delete, given that in many surface forms they remain intact (e.g., [tabl], [parl])? In particular, how does the child conclude that [tabl] and [tab] share the UR /tabl/ and are related through optional *L*-deletion but avoid concluding that [tabl] also shares the UR /tabl/ with [sabl](∼[sab]) 'sand' and that the two are related through an optional mapping of /t/ to [s]? Moreover, how does the child conclude that *L*-deletion is limited to word-final post-obstruent contexts, as stated in (1) and does not apply more freely? After all, as noted by Dell (1981), free (optional) *L*-deletion would be simpler to state, thus being favored by a learning criterion such as the evaluation metric of SPE (Chomsky & Halle 1968), and would never be counter-exemplified by the data.

*We thank Adam Albright, Naomi Feldman, Michael Kenstowicz, Donca Steriade, and the audiences at MIT and NELS 48.

[1]Where *L* stands for 'liquid'. Here and below we use phonological rules, which offer a particularly direct handle on the representation of both optionality and opacity, for our examples and our simulations.

[2]See Calamaro & Jarosz 2015 for motivation for focusing on distributional evidence in morpho-phonological learning.

Like optionality, opaque interactions pose a learning challenge by obscuring the form of a phonological process and its environment of application. Consider the case of counterfeeding opacity in Catalan, as discussed by Mascaró (1976) and Faust & Torres-Tamarit (2017) (for purposes of presentation we only consider here a simplified version of Catalan): one process, stated in (2a), deletes word-final nasals and appears to be obligatory (thus, the singular form of kuzín-s 'cousin.PL' is kuzí and not *kuzín); a second obligatory process, stated in (2b), deletes word-final obstruents following a nasal (e.g., kələ́n ~ kələ́nt-ə 'hot.MASC ~ hot.FEM').

(2) a. $N \rightarrow \emptyset$ /__# (obligatory)

b. $C \rightarrow \emptyset$ /N__# (obligatory)

Crucially, nasal deletion does not apply if the relevant nasal is word-final due to a following obstruent that deleted, which can be captured by ordering (2a) before (2b), an instance of counterfeeding opacity. The learning challenge that arises in this case concerns the process of nasal deletion. The child acquiring Catalan may observe, if they can make use of such so-called indirect negative evidence, that *NC#* sequences are absent in the surface forms of the input data and conclude that something like (2b) holds. Deciding that something like (2a) holds, on the other hand, seems more challenging: the presence of surface forms such as [kələ́n] can confuse a naive attempt to learn that word-final nasals delete.

Optionality and opacity are acquired by children (see, e.g., Dell 1981 and McCarthy 2007), but to date no learners in the literature have been shown to address this learning challenge from distributional evidence alone. For example, the learner of Boersma & Hayes (2001) has been shown to handle optionality (though not opacity) but requires access to URs and not just surface forms, which goes well beyond distributional evidence. Modeling the acquisition of opacity has received less attention in the literature, but the early learner of Johnson (1984) was shown to handle opacity (though not optionality); however, this learner, too, is not fully distributional, since it relies on analyzed paradigms. Finally, the learner of Calamaro & Jarosz (2015) acquires phonological processes from distributional evidence alone but does not handle either optionality or opacity.

The present paper provides a proof-of-concept demonstration of how both optionality and opacity can be acquired from distributional evidence alone. We do so using the principle of Minimum Description Length (MDL; Solomonoff 1964; Rissanen 1978), which provides a simple, unified framework for the acquisition of grammatical knowledge. MDL – and the closely related Bayesian approach to learning – have been helpful across a range of grammar induction tasks, in works such as Horning (1969), Berwick (1982), Ellison (1994), Rissanen & Ristad (1994), Stolcke (1994), Grünwald (1996), de Marcken (1996), Brent (1999), Clark (2001), and Goldsmith (2001), among others. Recently, Rasin & Katzir (2016) have used MDL to show how complete phonological grammars can be acquired distributionally within constraint-based phonology, and the present paper adapts that learner to rule-based phonology, which straightforwardly supports the representation of optionality

and opacity.[3] Section 2 uses the optionality challenge to motivate and informally present the MDL metric. Section 3 shows how the informal notions presented in section 2 can be made precise and turned into an actual learner. Section 4 illustrates the use of our MDL learner to acquire both the French optionality pattern and the Catalan opacity pattern from simplified toy corpora.

2. The MDL metric

The MDL metric balances two competing factors. The first factor is the simplicity of the grammar, written here as $|G|$, and usually measured in terms of how many bits are required to encode the grammar G according to a given formalism for stating grammars. The second factor is the fit of the grammar to the data, written here as $|D : G|$, where $D : G$ is the encoding of the data D given the grammar G (so that $|D : G|$ is the length of the encoding of D given G and, like $|G|$, is also measured in bits). The MDL metric balances the two factors by minimizing the sum of $|G|$ and $|D : G|$, as stated in (3).[4]

(3) 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 section 3 below we will make the MDL metric precise by showing how both $|G|$ and $|D : G|$ can be measured, using bits, in the case of a concrete representational framework for phonology. For the remainder of the present section, however, we will discuss (3) in terms of an informal understanding of the notions of grammar complexity and the choices needed to specify a surface form using the grammar.

Both components of the MDL metric – the size of the grammar, $|G|$, and the length of the encoding of the data given the grammar, $|D : G|$ – are crucial to ensure the right level of generalization, as we now briefly discuss. Minimizing the $|G|$ component, as in the evaluation metric of early generative grammar, favors grammars that are abstract and general: the shorter the grammar, the harder it is to encode within it *ad hoc* properties of the input data. In the case of optional *L*-deletion in French, minimizing $|G|$ will favor collapsing pairs such as [tabl] and [tab] onto a single UR (/tabl/) and deriving the two surface forms using a process of optional *L*-deletion. Stating the relevant process will lengthen the grammar somewhat, but if there are sufficiently many pairs like [tabl]~[tab] for which *L*-deletion makes it possible to store just one single UR, the savings obtained by the storing of just one UR for each relevant pair of alternating surface forms will more than justify the cost of stating the process of *L*-deletion. This amounts to generalization: if the learner now encounters a novel surface form such as [sabl], they will store it as /sabl/ and will correctly judge the *L*-deleted surface form [sab] to be licit even if they have not previously seen that form. However, if minimizing $|G|$ is the *entire* goal of the learner (as in the SPE metric but not as in MDL, where it is balanced by $|D : G|$), the learner will arrive at an overly

[3] See Katzir (2014) and Rasin & Katzir (2018) for reasons to think that MDL is a sensible framework for grammar learning.

[4] 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.

general hypothesis: they will not choose the correct form of *L*-deletion, as in (1), but rather an unrestricted form of *L*-deletion, as in (4).

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

The rule in (4) allows liquids to be deleted anywhere and not just between an obstruent and the end of a word, which is an incorrect result for French. However, it is shorter than the correct (1) and can account for the surface data, so it would incorrectly be preferred by a learning criterion that minimizes $|G|$ alone, a problem that was noted by Dell (1981).[5] In response to this problem, Dell advocated replacing the evaluation metric with a learning criterion that prefers the most restrictive grammar compatible with the data, a solution that came to be known as the *subset principle* (Berwick 1985; Wexler & Manzini 1987). In the present case of French optionality, the grammar with the unrestricted (4) generates a language that is a strict superset of the one with the restricted (1); consequently, the subset principle will reject the unrestricted (4) and choose the context-restricted (1), which is the correct result.

Minimizing the second component of the MDL metric, $|D : G|$, serves a similar purpose to the subset principle: a grammar that is more restrictive will allow the input data to be specified using fewer instructions than a less restrictive one and will thus typically result in a shorter $D : G$. For example, a grammar for French that uses the more restrictive (1) will require each surface occurrence of either [tabl] or [tab] to involve two specifications: first, the choice of the UR /tabl/ from the lexicon; and second, the specification of whether or not *L*-deletion has applied. A surface occurrence of [parl], on the other hand, will be easier to specify: the choice of UR is still required, but in this case it is not necessary to specify whether *L*-deletion applies since neither of the two liquids is in the relevant environment for deletion. Consider now an incorrect grammar for French that uses the less restrictive (4). In this grammar, *all* liquids are potential targets for deletion, and consequently each occurrence of a liquid requires a specification of whether *L*-deletion applies (in addition to the specification of the UR that was chosen). The specification of the data given the grammar, $D : G$, is consequently longer with the less restrictive (4) than with the more restrictive (1): the specification of the URs and that of whether *L*-deletion has applied for those liquids that appear between obstruents and word boundaries (as in [tabl] or [tab]) will be the same for both grammars, but the less restrictive grammar will also require specifying (spuriously) whether *L*-deletion has applied for occurrences of liquids that are in an inappropriate environment for deletion in actual French (as is the case for the two liquids in [parl]).

While choosing correctly between (4) and (1) in the setting just discussed, minimizing $|D : G|$ alone (or following the subset principle) gives rise to a problem of undergeneralization: it encourages the learner to adopt a complex grammar that memorizes as much of the input data as possible, which leads to overfitting. In the present case, 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 pairs differing only in whether they have a final post-obstruent liquid,

[5] Outside of phonology, this problem was discussed by Braine (1971) and Baker (1979).

we would expect them to adopt (1), even if for /sabl/ only one member of the pair has been observed so far. But if the learner is only trying to minimize $|D:G|$, this will not be possible: with (1), specifying each occurrence of [sabl] will require not just specifying the UR /sabl/ but also whether (1) has applied; this leads to a longer specification of $D:G$ – a greater $|D:G|$ – than that for an *ad hoc*, memorizing grammar that does not extend *L*-deletion to /sabl/ and for which only the choice of the UR /sabl/ needs to be specified. (The same can be restated in terms of the subset principle. For a grammar that includes (1), the language will include also the *L*-deleted form [sab], which makes the language a strict superset of the language of a memorizing grammar that does not extend beyond the observed data and does not license [sab].) In other words, the goal of minimizing $|D:G|$ alone blocks valid generalization.

We have seen that minimizing $|G|$ makes the child generalize but leads to overgeneralization, while minimizing $|D:G|$ protects from overgeneralization but leads to undergeneralization. It seems reasonable, then, to try to balance the two principles against each other: look for a relatively small G that allows for a relatively short $D:G$. This is exactly the idea behind MDL, as stated in (3) and adopted here. In the case of French *L*-deletion, minimizing the entire $|G|+|D:G|$ leads the learner to acquire *L*-deletion, as in the case of minimizing $|G|$ alone, but it ensures – thanks to the $|D:G|$ component – that the rule will be restricted to the right environment, as in (1), despite the slight additional cost in terms of $|G|$ compared to the overly general (4).[6] Likewise, while the $|D:G|$ component would benefit from a rule that does not delete *L* in /sabl/ (in the scenario discussed above), this incorrect result involves a complex grammar, and if the increase in $|G|$ is greater than the savings in $|D:G|$, the correct and more general (1) will be chosen.

The next section shows how the informal discussion above can be turned into an actual learner that can acquire not only optionality but other aspects of morpho-phonology including opaque rule interaction as in the Catalan example mentioned earlier.

3. Implementation

In order to turn the MDL evaluation metric into an actual phonological learner, we need to make explicit our representations. We do this in section 3.1, where we present the concrete representations we assume and the costs they induce in terms of MDL: $|G|$ is discussed in sections 3.1.1 (for the rules within G) and 3.1.2 (for the lexicon, which is likewise part of G), while $|D:G|$ is discussed in section 3.1.3. Section 3.2 completes the description of our learner by presenting the search procedure that we use to find a grammar that yields a good MDL score.

[6] Similarly, the $|D:G|$ component protects the learner from the incorrect potential generalization, mentioned in the introduction, in which [tabl] and [sabl] are collapsed onto one UR, /tabl/, with an optional rule mapping /t/ to [s]. This kind of generalization would presumably lead to a shorter $|G|$, since there will be fewer URs to store. However, it is not generally the case in French that forms in the language that have a [t] in them also have variants in the language that have an [s] instead (for example, the child might see forms like [aktif] or [partir], but they will not see forms like [aksif] or [parsir]). Consequently, specifying for each instance of /t/ whether it should change to [s] will generally be pointless, and the result will be an unnecessary increase to $|D:G|$ that far exceeds any possible gains in terms of $|G|$. This generalization, then, will correctly be dispreferred by the MDL metric.

3.1 Representations

3.1.1 Phonological rules

The general form of rules is as in (5), 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.

(5) *Rule format*

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

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 (which is more convenient for the purposes of the conversion to bits below) in (6b).

(6) *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 (7). 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.

(7) *Conversion table for rules*

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	...	...

Using the conversion table in (7), we can now encode the phonological rule of vowel harmony (in (6) above) by converting each element in the string representation in (6b) into bits according to (7) and concatenating the codes. To ensure unique readability, we use various delimiters to mark the end of the description of features, feature bundles, and the rule's components. The following is the result, and its length is 73 bits:

(8) *Vowel harmony rule (bit representation)*:

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

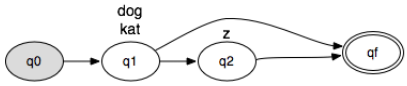
$\underbrace{0100}_{-}\underbrace{0110}_{cons}\underbrace{0000}_{\#_{f}}\underbrace{0100}_{-}\underbrace{1001}_{back}\underbrace{0010}_{\#_{rc}}\underbrace{1}_{1}\underbrace{0010}_{\#_{rc}}$

A phonological rule system is a sequence of phonological rules. Since 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 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.

3.1.2 Lexicon

The lexicon contains the URs of all the possible morphemes, along with information about their possible combinations. 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, for a toy version of English with two stems and one suffix, is provided in (9).

(9) *An HMM representation of a lexicon*



The HMM in (9) 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 (11) (derived from the concatenation of the string representations for the different states, as listed in (10); 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 the end of each state). We then convert the string to a bit-string using a conversion table, as in (12). As before, all choices at a given point are uniform, with the same code length for all possible selections at that point.

(10) *String representations of HMM states*

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	$q_2 q_f \#_S \text{z} \#_w \#_w$

(11) *String representation of an HMM*

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

(12) *Conversion table for HMM*

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
...	...

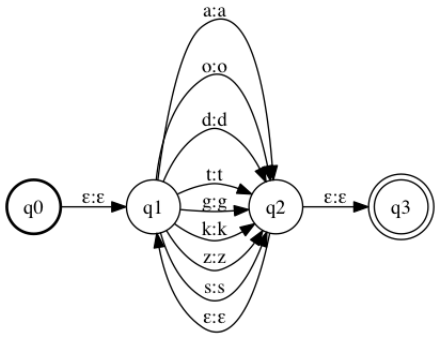
3.1.3 Data given the grammar

Given the grammar as described above, specifying a surface form involves: (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.

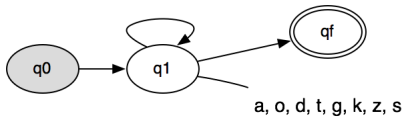
Our goal, given a surface form, is to determine the best way to derive it from the grammar in terms of code length. A naive approach to this parsing task would be to try all the ways to generate a surface form from the grammar. Even with simple grammars, however, this approach can be unfeasible. Instead,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

Let us illustrate the encoding of best derivations in the case of the form [kæts] using the FSTs for two simple grammars: one that treats all surface forms as arbitrary concatenations of segments from a basic inventory (with no complex morphemes and no phonological rules) and another that has the morphemes /kæt/, /dɔg/, and /-z/ in the lexicon, along with a rule of voicing assimilation. First, consider the FST in (13), which corresponds to a grammar with the lexicon in (14) and no phonological rules. Using this FST, encoding the word [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 (16), is in (15).

(13) *Naive FST*



(14) *Lexicon corresponding to the naive FST*



(15) *Encoding of a surface form using the naive FST*

100	0	000	0	010	1	110	1
k	$q_2 \rightarrow q_1$	**a**	$q_2 \rightarrow q_1$	**t**	$q_2 \rightarrow q_1$	**s**	$q_2 \rightarrow q_3$

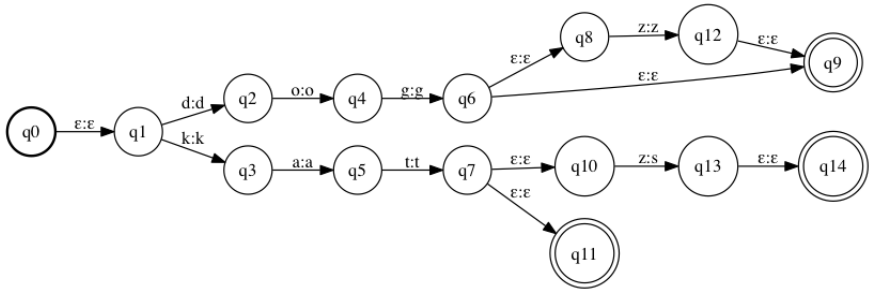
(16) *Conversion table for naive FST*

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

(17) *A more complex FST*



(18) *Encoding of a surface form using the more complex FST*

0	0
$q_1 \rightarrow q_3$	$q_7 \rightarrow q_{10}$

Consider now the more complex FST in (17), which corresponds to a grammar with the lexicon in (9) and the English voicing assimilation rule. This FST corresponds to a more restrictive grammar: differently from the simpler FST in (13), the present FST can only generate a handful of surface forms. Consequently, the present FST offers a shorter *D*:*G*. Specifically, since specifying [kæts] requires 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 (18).

3.2 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 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.

4. Simulations

The present section provides simulations in which the MDL learner described in section 3 is faced with unanalyzed data exhibiting two linguistically-relevant patterns.[7] Section 4.1 illustrates our learner's acquisition of optionality, using a dataset based on the case of optional French *L*-deletion discussed above. Section 4.2 shows that the MDL learner succeeds on a case of counterfeeding opacity modeled after the case of Catalan.

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 the phonological pattern under consideration.

4.1 Optionality

Our first dataset shows a pattern modeled after French *L*-deletion (Dell, 1981), as discussed above. Recall that the challenge for the learner is to strike the right balance between economy and tightness of fit to the data. 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 104 words, including 36 collapsible pairs (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 (19). The encoding length of the data given the grammar was multiplied by 50 and the encoding length of the HMM was multiplied by 20.

(19) tab, tabl, arb, arbr, kapab, kapabl, parl, partir, final, aktif, …

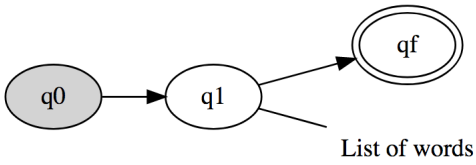
[7] The code for the learner is available at https://github.com/taucompling/morphophonology_spe.

(20) *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/).*

$G_{final} =$ {

RULES: $R_1 : [+liquid] \rightarrow \emptyset / [-son]\_\_(optional)$

LEX:



List of words = tabl, arbr, kapabl, parl, partir, final, aktif,...

}

Description length: $|G_{final}| + |D{:}G_{final}| = 40,113 + 40,200 = 80,313$

The learner induced the correct optional rule and converged on the target lexicon in (20). 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:

(21) a. *Correct Hypothesis*

- $R_1 : [+liquid] \rightarrow \emptyset / [-son]\_\_$ (optional)
- Description length: $|G| + |D{:}G| = 40,113 + 40,200 = 80,313$

b. *Over-generating Hypothesis*

- $R_1 : [+liquid] \rightarrow \emptyset / \_\_$ (optional)
- Description length: $|G| + |D{:}G| = 40,105 + 43,200 = 83,305$

4.2 Counterfeeding opacity

Our next dataset was designed to test the learner on the problem of counterfeeding opacity. Recall from the introduction that in Catalan (Mascaró 1976, Faust & Torres-Tamarit 2017), nasals are deleted word-finally (22a) and a rule of cluster simplification deletes a stop word-finally after a nasal (22b) and creates the environment for final-nasal deletion, which does not apply on the surface in (22b).

(22) a. kuzí ~ kuzín̲-s ‘cousin.SG ~ cousin.PL’

b. kəlén ~ kəlén̲t-ə ‘hot.MASC ~ hot.FEM’

To generate our corpus, 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 and applying final-nasal deletion and cluster simplification, in this order ((23), repeated from (2)), and we multiplied $|D : G|$ by 10. The stems and suffixes were taken from a Catalan dictionary. A sample of the data is given in (24). The learner converged on the expected lexicon and on the two rules – final-nasal deletion (23a) and cluster simplification (23b) – and their correct ordering, as in (25).

(23) a. $N \rightarrow \emptyset$ /__# (obligatory)

b. $C \rightarrow \emptyset$ /N__# (obligatory)

(24)

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

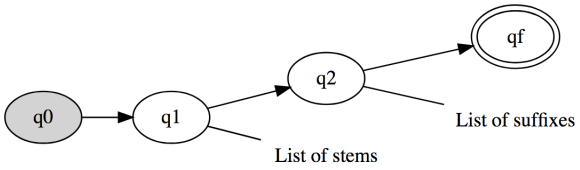
(25) *Final grammar for the counterfeeding opacity simulation. The grammar includes final-nasal deletion and cluster simplification (in this order) and a segmented lexicon. Word boundaries are represented directly using the symbol '#'.*

$G_{final} =$

RULES: $R_1 : [+nasal] \rightarrow \emptyset /$ __ #

$R_2 : [-cont] \rightarrow \emptyset / [+nasal]$ __ #

LEX:



List of stems = kuzin, plan, kalent, blank, kasa, metal, kap, ...

List of suffixes = #, et#, a#, ik#, s#, ...

Description length: $|G_{final}| + |D{:}G_{final}| = 495 + 4,390 = 4,885$

5. Summary

We presented an MDL-based learner for the unsupervised joint learning of lexicon, morphological segmentation, and ordered phonological rules from unanalyzed surface forms. The current learner goes beyond the literature in two main respects. First, it can handle systems that involve not just obligatory processes but also optional ones. And second, it can handle a case of opaque rule interaction. In addition to being the first learner we are familiar with that accomplishes these tasks, it is worth noting that the learner does so using a simple, general learning criterion and representations (namely, context-sensitive rewrite rules) that have been 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.

References

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

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

Chomsky, Noam, & 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.

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

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

Faust, Noam, & Francesc Torres-Tamarit. 2017. Stress and final /*n*/ deletion in Catalan: Combining strict CV and OT. *Glossa: a journal of general linguistics* 2.

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

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 & 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*. University of Michigan Press.

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., & 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.

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

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

McCarthy, John J. 2007. Derivations and levels of representation. In *The Cambridge*

handbook of phonology, ed. Paul de Lacy, 99–117. Cambridge University Press.
Rasin, Ezer, & Roni Katzir. 2016. On evaluation metrics in Optimality Theory. *Linguistic Inquiry* 47:235–282.
Rasin, Ezer, & Roni Katzir. 2018. A conditional learnability argument for constraints on underlying representations. Ms., MIT and TAU, March 2018.
Rissanen, Jorma. 1978. Modeling by shortest data description. *Automatica* 14:465–471.
Rissanen, Jorma, & 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.
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.
Wexler, Kenneth, & Rita M. Manzini. 1987. Parameters and learnability in binding theory. In *Parameter setting*, ed. Thomas Roeper & Edwin Williams, 41–76. Dordrecht, The Netherlands: D. Reidel Publishing Company.

Ezer Rasin, Iddo Berger, Nur Lan, Roni Katzir
rasin@mit.edu, iddoberger@gmail.com, nurxlan@gmail.com, rkatzir@post.tau.ac.il