To appear in the Proceedings of NELS43.

When Worst is Best: Grammar Construction in Phonological Learning[*]

Bruce Tesar

Rutgers University, New Brunswick

1. Introduction

This paper explores a benefit of a language learner that accumulates and stores selected information about the target grammar, separate from the representation of the grammar itself. The benefit of interest is the ability to use the accumulated information to construct grammars other than the most plausible "best guess" grammar, and to use those grammars to advance learning. In particular, there are circumstances where constructing the "worst" grammar consistent with accumulated information actually allows the learner to progress more efficiently.

2. Traditional error-driven learning

Traditional error-driven learning posits that a learner, at any given time, has exactly one grammar represented. That grammar constitutes the learner's "best guess" for the target grammar at that point in time. The learner processes data one word at a time.[1] When processing a word, the learner attempts to parse the word using its current grammar. If parsing fails, an error has been detected: the failure to parse the word is taken as an indication that the learner's current grammar is incorrect. In the event of an error, the learner attempts to modify its grammar, producing a new "best guess" grammar. Once the learner adopts the target grammar as its grammar, no further errors should be detected, and the learner will not attempt to further modify its grammar. The term "error-driven learning" is due to Wexler and Culicover (1980), but this approach to grammar learning dates back to the Learnability in the Limit framework (Gold 1967), and the idea of error-

[*] I wish to acknowledge helpful discussions with Alan Prince, the audience of the 3rd North East Computational Phonology Circle (MIT 2009), and the Rutgers Optimality Reading Group. Any errors are the sole responsibility of the author.

[1] For concreteness, I will describe learners of word-level phonology, but concepts like error-driven learning apply more generally.

driven hypothesis updating can be traced back to early work in machine learning (Rosenblatt 1958). Examples of this approach in contemporary phonological learning include Error-Driven Constraint Demotion (Tesar and Smolensky 2000), the Gradual Learning Algorithm with stochastic Optimality Theory (Boersma 1998), and the Gradual Learning Algorithm with Harmonic Grammar (Pater 2008).

Importantly, the "best guess" grammar is the only information retained by the learner as it confronts further data. If a word happens to be generated by the learner's current grammar, no representation is made of which parts of the current grammar were essential for the successful generation of that word. This means that if other data cause the learner to change to a different grammar, there is nothing to force future grammars adopted by the learner to correctly generate that word. More generally, the learner does not retain any distinction between the parts of their current grammar that were motivated by past data and the parts that are merely in place by default. As will be illustrated below, the ability to distinguish the *known* part of the "best guess" grammar from the *guess* part of the "best grammar" can be extremely valuable, especially in the learning of phonological underlying forms.

3. Information Accumulation

3.1 Obtaining and storing information about the target grammar

Learners that retain only a single current grammar are minimal in their memory storage demands: any learner must store a grammar, these learners store nothing else. The alternative considered here is for the learner to store information apart from any particular grammar, information that can then be used by the learner to generate a grammar when needed. The result is a subtle but significant change in how the learner responds to a newly observed piece of data. Instead of asking if the observed word causes an error with respect to their current grammar, the learner asks if the observed word can provide any new information regarding the target grammar (information not already entailed by the learner's stored information).

Several capacities are required by an information accumulation approach, including the capacity to store information constructed from previous data, to efficiently generate a grammar when needed, to efficiently evaluate a new word *with respect to the learner's currently stored information*, and to extract and store new information entailed by the new word.

An example of the information accumulation approach in contemporary phonological learning is MultiRecursive Constraint Demotion, abbreviated MRCD (Tesar 2004). MRCD is an algorithm for the learning of constraint rankings from data. Like Error-Driven Constraint Demotion (EDCD) and the Gradual Learning Algorithm (the GLA), MRCD adopts Optimality Theory (Prince and Smolensky 2004) as its core linguistic theory, and it operates in an on-line fashion, processing each word as it is encountered. Also like EDCD and the GLA, MRCD represents information newly obtained from a word with a *winner-loser pair* (Tesar and Smolensky 2000). The crucial difference lies in what is done with a constructed winner-loser pair. EDCD and the GLA are traditional

error-driven learners: they use the single winner-loser pair to directly modify their current grammar (a stratified hierarchy for EDCD, ranking values for the GLA) and then discard the winner-loser pair. MRCD instead retains a list of all the winner-loser pairs it has constructed. Whenever a new winner-loser pair is added to the list, MRCD constructs a new grammar (a stratified hierarchy) based on that list (but not on the prior grammar).

3.2 Phonotactic learning

Phonotactic learning is the problem of finding the most restrictive grammar consistent with a set of surface forms (Hayes 2004, Prince and Tesar 2004); for recent analysis and discussion see (Magri 2012). The learner, lacking initial knowledge of the underlying forms, is presumed to adopt, for each observed word, a hypothesized phonological input that is identical to the observed surface form of the word. The learner's goal is to arrive at a constraint ranking that maps each observed word (as input) to itself (as surface form), but that generates as few other surface forms as possible.

MRCD has been applied to phonotactic learning, in the form of the Biased Constraint Demotion (BCD) algorithm (Prince and Tesar 2004). BCD enforces restrictiveness by selecting, from among the rankings consistent with its list of winner-loser pairs, the ranking that has the most markedness constraints dominating the most faithfulness constraints; I will call this the FaithLow ranking for a list of winner-loser pairs. When BCD processes a word, it constructs an input that is identical to the surface form of the word, and then generates the candidate for that input that is optimal with respect to its current grammar. If the optimal candidate does not match the observed word, then a winner-loser pair is formed, with the observed word as the winner, and the generated candidate as the loser, and that pair is added to the learner's list of winner-loser pairs. BCD then generates a new constraint hierarchy from the (now longer) list of pairs. Because the loser was optimal with respect to the previous ranking, the winner-loser pair will contain information that the learner did not previously possess. Once the learner has accumulated a sufficient set of winner-loser pairs, the generated hierarchy will correctly map all observed words to themselves, and no further winner-loser pairs will be constructed; phonotactic learning is at that point complete.

As just described, BCD is similar to traditional error-driven learning in some respects. It uses its current "best guess" grammar to parse a word, and only changes its grammar if an error occurs. However, the information accumulation nature of BCD allows for a different interpretation of the same algorithm. Because the learner reconstructs a grammar based on its accumulated information, it can construct different grammars for different purposes from the same information. The learner, in pursuing the goal of obtaining new information from a word, constructs the grammar (consistent with its stored information) that is most likely[2] to result in an error on the observed word. Errors result in the construction of new winner-loser pairs, which constitute new information about the target grammar. The grammar most likely to result in an error is the grammar

[2] The phrases "most likely" and "least likely" are used rather loosely in this paper, and should not be interpreted as denoting any rigorously quantified probabilistic notion.

least likely to be compatible with the observed word. In the present circumstance, that grammar is the most restrictive one: the one that generates as few words as possible.

In the case of phonotactic learning, the "best guess" grammar (the most restrictive grammar consistent with the learner's ranking information) and the "most likely to cause an error" grammar are one and the same. In other stages of learning, however, the two are distinct. The rest of this paper argues that, in those stages, it is the grammar most likely to cause an error that the learner should construct when processing a word, even when that grammar is the "worst" in the sense of being the one that the learner would regard as *least* likely given the data it has seen. This strategy of constructing different grammars for different learning purposes from the same information is made possible by the information accumulation approach to learning.

4. Using the "worst" underlying form

The information accumulation approach can be extended to the simultaneous learning of constraint rankings and phonological underlying forms (Merchant 2008, Tesar 2006); see also (Akers 2012). This kind of learner stores both a list of winner-loser pairs (information about the ranking) and a lexicon of (partial) underlying forms for observed morphemes (information about the underlying forms), as well as a list of surface forms for words it has observed and analyzed (one surface form per word). Both the ranking information and the underlying form information are accumulated over time, across observed forms, and are used to generate a hypothesized grammar whenever necessary.

4.1 Setting features with inconsistency detection

The values of some underlying features can be set using inconsistency detection (Merchant 2008, Merchant and Tesar 2008). The underlying value of a feature can be set based on a single word if it is the case that all possible underlying forms with different values for the feature can be shown to be inconsistent for that word (will not succeed given the learner's current information about the ranking).

As a simple example, consider the word [paká:], with root [pa] and suffix [ká:], in a linguistic system in which a vowel has two binary features[3] specified underlyingly: stress (accent) and length. Each morpheme (the root and the suffix) has one vowel and so two underlying features, thus there are 16 possible inputs for the word. Focusing on the length feature of the suffix, it will have the value +long in 8 of the possible inputs, and it will have the value –long in the other 8 inputs. Given that the word has been observed, at least one of the possible inputs must map to the word's surface form [paká:]. If all of the inputs with the value of –long for the suffix vowel can be shown to be inconsistent with the learner's information about the constraint ranking, then the correct input must have the value of +long for the suffix vowel, and the learner can set that value for the suffix in the lexicon (where it will equally apply to any word containing that suffix).

[3] The term "feature" is here used rather generically, intended to apply equally to traditional segmental features and suprasegmental elements.

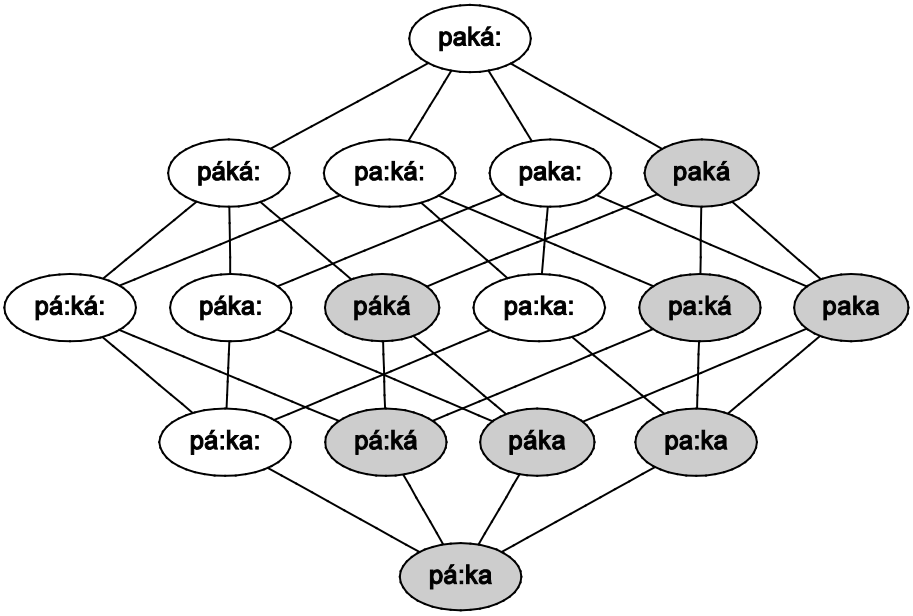
4.2 Output-driven maps

The simplest, most general way of setting a feature using inconsistency detection is to construct all of the relevant inputs and evaluate them for consistency. However, this can be computed much more efficiently if the learner can assume that the target grammar realizes an output-driven map (Tesar 2011, Tesar in press).

A map from phonological inputs to grammatical candidates is said to be *output-driven* if, for every grammatical candidate /A/→[X] of the map, if input /B/ has greater similarity to output [X] than /A/ has, then /B/ also maps to [X]. Similarity here is relational: input /B/ can only be said to have greater similarity to [X] than /A/ has if there is a candidate /B/→[X] that has only a subset of the input-output disparities that candidate /A/→[X] has.

Output-drivenness imposes structure on the space of inputs for a surface form. This structure normally takes the form of a lattice, as illustrated in the figure in (1).

(1) The relative similarity lattice for output [paká:] (greater similarity is higher).



For the moment, ignore the shading on some of the nodes of the lattice. Each node represents a candidate with output [paká:], a two-syllable word where the second syllable contains a stressed, long vowel. Each node is labeled with the input of the candidate it represents. The edges of the lattice are relative similarity relations between candidates; for a given candidate, the candidates with greater similarity are the ones that are above it in the lattice. Note that the top node of the lattice, paká:, is the one with an input that is identical to the output; identity is as similar as input and output can get.

In an output-driven map, the candidates in a lattice have an entailment relation with respect to grammaticality: if a given candidate is grammatical, then every candidate above it in the lattice is necessarily also grammatical. In figure (1), the node labeled páka:

represents the candidate /páka:/→[paká:]. If that candidate is grammatical, then the three candidates above it are also grammatical, that is, those three inputs also map to the *same* output, [paká:].

The structure of output-driven maps can be exploited to great effect in learning. Because grammaticality entails upward in the lattice, ungrammaticality entails downward in the lattice: if a candidate is not grammatical, then no candidate below it in the lattice is grammatical either. Thus, if a given candidate is inconsistent with the learner's ranking information, and thus cannot be grammatical, the learner may safely conclude that all candidates lower in the lattice are also inconsistent, without needing to explicitly construct and evaluate them.

If a map is output-driven, then the identity map property (Prince and Tesar 2004) is an automatic consequence: any grammatical form maps to itself. In figure (1), the identity candidate for [paká:] is the top of the lattice: if any input maps to [paká:], then /paká:/ does. This automatically captures the justification for using an input identical to the output in phonotactic learning: if [paká:] is the observed output for a word, then the target grammar must map /paká:/ to [paká:], whether or not /paká:/ is the correct input for that particular word.

The highest shaded node in figure (1) represents a candidate, /paká/→[paká:], with only one input-output disparity: the length of the suffix vowel. Put another way, the input differs from the output only in the value of the length feature of the suffix: it is –long in the input. Every other candidate with –long for the suffix in the input has at least one other disparity as well, and is ordered below /paká/→[paká:] in the lattice. The candidates with suffix input –long are shaded in figure (1), and form a sublattice. The key observation is that if /paká/ cannot map to [paká:] in the target grammar, then neither can any other candidate with suffix input –long. The single candidate /paká/→[paká:] can successfully serve as a proxy for the entire sublattice of suffix input –long candidates, and the learner need only construct and evaluate that single candidate for consistency with the learner's ranking information. If that candidate is inconsistent, then the learner will set the length feature of the lexical entry of the suffix to +long.

The analogous property holds for every other underlying feature: all of the candidates with the same underlying value for a particular feature form a sublattice, and the learner can determine if the feature can be set based on the word by constructing and evaluating only the single candidate at the top of the sublattice for the inputs with the value opposite the feature's surface realization. With respect to the lattice in figure (1), the second row (from the top), which contains 4 candidates, are the only candidates that the learner needs to evaluate for consistency: they will tell the learner everything that can be learned about the lexicon via inconsistency from that word (at that point in learning).

For extensive further discussion of output-driven maps, see (Tesar in press).

4.3 Input subspace evaluation

As just shown, the structure of output-driven maps means that, given N unset features in a word, the learner can learn what there is to learn from the word at that point by evaluating only candidates for N inputs, even though there are candidates for 2^N inputs

available. However, there is a further improvement that can be implemented, one also made possible by the structure of output-driven maps.

In figure (1), the bottom node of the lattice differs from the surface form on every feature; it is the input that is "least similar" to the output. It is also below every other node in the lattice. If that candidate is consistent with the learner's ranking information, then every other candidate is necessarily also consistent. If all candidates are consistent, there is clearly no point in attempting to set individual feature values for that word, as no inconsistencies will be detected. If the bottom candidate is inconsistent, that fact does not guarantee that the learner will be able to set a feature, but it is a useful pretest, as it only involves the construction and evaluation of one candidate. This extends to cases where some features have been set, but not others: the set of viable candidates (those with inputs consistent with the set features of the learner's lexicon) will form a sublattice, with a single bottom candidate.

Evaluating the bottom viable candidate for a word, then, functions similarly to error detection (at least for purposes of learning underlying feature values): if the bottom candidate is consistent with the learner's grammatical information, then the learner concludes there is nothing to be learned from the word, while if it is inconsistent, then the learner will invest further effort in an attempt to learn new information from the word. An interesting divergence, however, is that the bottom viable candidate represents the "worst guess" candidate for the input for the word, in a sense. If called upon to select the most plausible input for the word, absent evidence to the contrary, it would make sense for the learner to select input feature values that match their surface realizations in the word. But in this case, the most informative input to evaluate is the one that mismatches the surface values as much as possible, that is, the one containing the "worst" choice of underlying feature values.

The computational benefit of this initial word evaluation will be particularly great when the target language contains a number of non-contrastive features. Underlying features that are non-contrastive will not be set by an inconsistency-based learner, because by definition either value of the underlying feature will be consistent. A learner which blindly attempts to set every unset feature for every word it processes could engage in a significant amount of unproductive computation for words that contain no new information but that contain multiple instances of non-contrastive (hence unset) features. For such words, initial word evaluation will test the input at the bottom of the viable sublattice, determine that it is consistent, and move on, avoiding repeated futile efforts to set unsettable features.

The learner proposed here does not employ the "worst" choice simply to be perverse, and it has not concluded that selecting underlying feature values that mismatch the surface realizations yields to most plausible lexical hypothesis. It is constructing a candidate with those values precisely because that candidate is the most likely to be wrong, the most likely to prove inconsistent with the learner's ranking information. This strategy is available to the learner because it doesn't simply store a single, fully specified "best guess" grammar (ranking and lexicon), but instead accumulates information about the target grammar: winner-loser pairs constitute information about the ranking, and a partially specified lexicon with some features set and others not constitutes information

about the target grammar's lexicon. By storing this information, the learner is capable not only of generating a "best guess" grammar for purposes of language use, but is also capable of generating other grammars for learning purposes, including the "worst" underlying forms.

5. Using the "worst" ranking

5.1 Non-phonotactic ranking information

Section 3.2 discussed the learning of phonotactic ranking information, where a learner parses an identity candidate for an observed word with respect to the FaithLow ranking (the constraint hierarchy with the faithfulness constraints ranked as low as possible). If the identity candidate is not optimal, then the learner constructs a winner-loser pair and generates a new constraint hierarchy. A standard motivation for using the FaithLow ranking is that it is the most restrictive ranking, and therefore the most plausible ranking given the information the learner possesses. However, as mentioned in Section 3.2, a different motivation can be provided for using the same ranking: the FaithLow ranking, because it is the most restrictive, is the most likely to be inconsistent with data yet to be observed.

For the learning of phonotactic ranking information, the two motivations converge on the same constraint hierarchy. The two diverge, however, when it comes to the learning of non-phonotactic ranking information. Non-phonotactic ranking information is information specific to non-faithful mappings, where the input is *not* identical to the output. Learning non-phonotactic ranking information is greatly aided by a learner having confidence in the underlying value of at least one feature such that it surfaces non-faithfully in a word. The Output-Driven Learner (ODL) interleaves the learning of underlying feature values with the learning of non-phonotactic ranking information: whenever an underlying feature is set for a morpheme, the learner looks for a word containing that morpheme in which the newly set feature surfaces unfaithfully, and attempts to obtain new non-phonotactic ranking information from it.

The algorithm used by the ODL to obtain non-phonotactic ranking information from a word is quite similar to that used in phonotactic learning. The ODL constructs a constraint hierarchy, constructs a candidate for the observed word by adopting values for *unset* features that are identical to their surface realizations in the word, and determines if the candidate is optimal for the constructed grammar. If the candidate is not optimal, then a winner-loser pair is constructed and added to the learner's list. The difference lies in the constraint hierarchy that is constructed for purposes of evaluating the candidate.

5.2 The MarkLow ranking bias

The ODL pursues non-phonotactic ranking information by constructing, based on its list of winner-loser pairs, a constraint hierarchy in which the markedness constraints are ranked as low as possible, which I will call the MarkLow ranking. The MarkLow ranking

is constructed using the same procedure as for the FaithLow ranking, only with the roles of markedness and faithfulness constraints reversed.

The ODL generates the MarkLow ranking because that ranking is less likely to generate the constructed candidate. The candidate necessarily includes features which are not faithfully realized, meaning that the faithfulness constraints sensitive to those features must be violated, likely as a result of domination by one or more markedness constraints. By ranking the markedness constraints as low as possible, the MarkLow ranking reduces the chances of the relevant faithfulness constraints being dominated by the necessary markedness constraints. If, as a consequence, the constructed candidate is not optimal, then the resulting winner-loser pair will include information indicating that the faithfulness constraints must be dominated, and that new ranking information will limit the learner's future constructed grammars appropriately.

Again, the learner does not regard the MarkLow ranking as the "best guess" grammar. The restrictiveness issue has not gone away (and it never will). If called upon to generate the most plausible ranking, the learner would generate the FaithLow ranking. But, because it is employing an information accumulation approach, storing a list of winner-loser pairs as information about the target ranking, the learner can generate different rankings for different purposes. The MarkLow ranking proves useful in the pursuit of non-phonotactic ranking information, despite the fact that it is the "worst guess" given the learner's information about the target ranking.

5.3 Learning non-phonotactic ranking information

Section 4.2 showed how the length feature for a suffix was set to +long by the ODL. Here, I will continue that illustration by showing how the ODL uses that newly set underlying feature to obtain non-phonotactic ranking information. This illustration comes from a learning simulation on a language from a system of six constraints. The markedness constraints are listed in (2), and the faithfulness constraints are listed in (3).

(2) MAINLEFT – the initial syllable should be stressed (McCarthy and Prince 1993)
MAINRIGHT – the final syllable should be stressed (McCarthy and Prince 1993)
WSP – heavy syllables (here, long vowels) should be stressed (Prince 1990)
NOLONG – vowels should not be long on the surface (Rosenthall 1994)

(3) ID[stress] – IO correspondents should agree in stress (McCarthy and Prince 1995)
ID[long] – IO correspondents should agree in length (McCarthy and Prince 1995)

The learner had initially performed phonotactic learning, yielding ranking information equivalent to the relations listed in (4) and (5). The information in (4) reflects that the language exhibits contrastive stress. The information in (5) reflects that long vowels can appear on the surface in the language.

(4) ID[stress] ≫ {MAINLEFT,MAINRIGHT}

(5) ID[long] ≫ NoLong

Recall that the learner had just set the underlying length of the suffix to +long in section 4.2. When the learner sets an underlying feature for a morpheme, it checks to see if it knows of any words containing the same morpheme where the newly set feature surfaces with the opposite value. In this case, the learner has the surface form [páka] for a different word containing the same suffix, but where the suffix surfaces as short (–long) and unstressed (–stress). Because the learner has only set the length feature of the suffix at this point, it must construct a fully specified input for evaluation purposes. The top of the sublattice of viable inputs for surface form [páka] is /páka:/, with all of the features matching their surface realization in the word except for the suffix length. One of the viable inputs must be correct, and if any of them maps to the output, then the top one does. Thus, the learner will evaluate the candidate /páka:/→[páka].

We want to examine the consequences of evaluating this candidate with each of two different rankings: FaithLow and MarkLow. The FaithLow ranking, shown in (6), is the "best guess" grammar at this point.

(6) WSP ≫ ID[stress] ≫ {MainLeft,MainRight} ≫ ID[long] ≫ NoLong

With the FaithLow ranking, WSP dominates ID[long], not due to requirements stored in the list of winner-loser pairs, but by default because of the ranking bias inherent in FaithLow. As a result, the candidate selected for evaluation, /páka:/→[páka], is in fact optimal with this ranking, and so no new information is obtained.

The MarkLow ranking, shown in (7), is the "worst guess" grammar, the one that is likely to be least restrictive while consistent with the learner's current ranking information.

(7) {ID[stress], ID[long]} ≫ {WSP, MainLeft,MainRight,NoLong}

With the MarkLow ranking, the WSP is dominated by ID[long], again not due to requirements stored in the list of winner-loser pairs, but by default because of the ranking bias inherent in MarkLow. But as a result, the candidate selected for evaluation, /páka:/→[páka], is *not* optimal. The dominant ranking of ID[long] ensures that an underlyingly long vowel will surface as long, even if the vowel is unstressed on the surface, so the optimal output is [páka:]. The learner now creates a new winner-loser pair, based on input /páka:/, with [páka] as the winner's output and [páka:] as the loser's output. The ERC for the winner-loser pair is shown in the tableau in (8), in the row labeled *new*. On its own, this winner-loser pair indicates that at least one of WSP and NoLong must dominate ID[long], and thus requires that ID[long] must be dominated. In combination with some of the phonotactic ranking information, represented by the ERC in the row labeled *phonotactic*, the learner is able to determine that WSP must dominate both ID[long] and NoLong. The result of the combination of the two is revealed by taking the fusion of the two ERCs (Prince 2002), shown in the bottom row of (8), labeled *fusion*.

(8) The ERC for the new winner-loser pair, along with one of the phonotactic pairs.

	WSP	MAINLEFT	MAINRIGHT	NOLONG	ID[stress]	ID[long]
phonotactic				L		W
new	W			W		L
fusion	W			L		L

The newly obtained ranking information is clearly non-phonotactic ranking information: it concerns the necessary domination of a faithfulness constraint, ID[long], by another constraint, WSP. That information indicates that the target grammar would rather shorten an underlyingly long vowel than faithfully realize an underlyingly long vowel in an unstressed (on the surface) syllable. Non-phonotactic ranking information requires evidence of unfaithful mappings, which is obtained by a learner setting the underlying value of a feature based on a word in which the feature must be faithfully realized, and then finding a different word in which the feature is unfaithfully realized.

The new ranking information was obtained in part by determining that the target grammar must generate the unfaithful mapping /páka:/→[páka], with shortening of the suffix vowel. However, that mapping only provides new ranking information if the learner's current information permits at least one grammar in which /páka:/ is mapped to something other than [páka]. To represent new ranking information as a winner-loser pair, the learner must identify a specific candidate mapping /páka:/ to something else such that the candidate is optimal for at least one currently viable grammar. The ODL increases its chances of finding such a candidate by constructing the MarkLow ranking. With the markedness constraints ranked as low as possible while remaining consistent with the learner's current (at that point) ranking information, the relevant faithfulness constraint, ID[long], is more likely to *not* be dominated by relevant markedness constraints, and the ranking is more likely to faithfully preserve the length feature of the suffix, thus providing an informative losing candidate for learning. In this example, the MarkLow ranking provides the key losing candidate, despite being the "worst guess" ranking, while the "best guess" ranking, the FaithLow ranking, does not. For purposes of obtaining non-phonotactic ranking information, the "worst" ranking is the best one to use.

What is the significance of the new ranking information? Once the new winner-loser pair has been added to the learner's list, if called upon to generate its best guess grammar, the learner will generate a new FaithLow ranking, to enforce restrictiveness. That ranking is the one in (6), exactly the same ranking as before it obtained the new winner-loser pair. At that instant in learning, the new information has not altered the learner's best guess grammar. The impact could be more significant as learning progresses further, however. The new ranking information explicitly rules out grammars that were previously viable (in particular, grammars that allow long, unstressed vowels on the surface). The more explicit ranking information that a learner represents, the more opportunities there are for the learner to detect inconsistency when attempting to learn from other forms of the language, such as when attempting to set the underlying values for other features.

6. Conclusion

The Output-Driven Learner is an example of a learner based on information accumulation. The ODL accumulates winner-loser pairs as information about the ranking of the target grammar, and accumulates set values for features of underlying forms as information about the lexicon of the target grammar. By storing such information, rather than storing only a single hypothesis grammar, the learner has greater flexibility: it can generate different grammars consistent with its stored information for different purposes.

Presumably, when the learner is called upon to engage in standard language use, it will make use of its "best guess" grammar, based on its stored information. The learner is still predicted to have a current grammar at any point during the course of learning, and that grammar is predicted to reflect what the learner has learned up to that point.

Retaining the stored information about the target grammar, separate from the learner's "best guess" grammar, has significant advantages for learning. By retaining accumulated information, the learner is able to distinguish between aspects of their current grammar that are directly motivated by earlier data, and aspects of the their current grammar that are present by default, out of the need to fill in gaps in order to have a functioning grammar. The learner can distinguish between the *known* part and the *guess* part. Such a distinction is necessary if a learner is going to productively use inconsistency detection in combination with data, because the learner needs to be able to distinguish between grammars that are consistent with previously seen data (viable grammars) and grammars that are inconsistent with previously seen data.

In the learning of underlying forms, the information motivated by previously observed data takes the form of set features. Instead of producing a single hypothesis for the input of a word, and asking if it currently works, the learner asks if the word can be used to set any currently unset features. By virtue of the structure of output-driven maps, the learner can compute this efficiently by evaluating the "worst" guess input for the word. If the worst guess input works for the word, the learner knows there is nothing to learn from the word at this point. Put another way, if the worst guess succeeds, the learner knows that the success is entirely due to the known part constraining the worst guess, and not due to the guess part.

The ability to generate grammars from separate stored information creates the opportunity for a learner to construct different grammars from the same stored information for different purposes in learning. This is useful when obtaining ranking information, because the standard techniques for obtaining ranking information involve using a constraint hierarchy to evaluate a candidate. New ranking information is obtained when an existing hierarchy fails to generate a candidate that should be grammatical. Thus, new ranking information is more likely to be obtained when a constraint hierarchy is constructed that, while consistent with the learner's stored information, is more likely to fail to generate the desired candidate. For the learning of phonotactic ranking information, the best guess hierarchy, the FaithLow ranking, is also the most likely to produce an informative error. But for the learning of non-phonotactic ranking information, it is the worst guess hierarchy, the MarkLow ranking, that is more likely to

produce an informative error. For some learning tasks, the "worst" grammar works the best.

References

Akers, Crystal. 2012. Simultaneous Learning of Hidden Structures by the Commitment-Based Learner. Doctoral dissertation, Rutgers University, New Brunswick.

Boersma, Paul. 1998. *Functional Phonology*. The Hague: Holland Academic Graphics.

Gold, E. Mark. 1967. Language identification in the limit. *Information and Control* 10:447-474.

Hayes, Bruce. 2004. Phonological acquisition in Optimality Theory: The early stages. In *Constraints in Phonological Acquisition*, eds. René Kager, Joe Pater and Wim Zonneveld, 158-203. Cambridge: Cambridge University Press.

Magri, Giorgio. 2012. Convergence of error-driven ranking algorithms. *Phonology* 29:213-269.

McCarthy, John J., and Prince, Alan. 1993. Generalized alignment. In *Yearbook of Morphology*, eds. Geert Booij and Jaap Van Marle, 79-154. Dordrecht: Kluwer.

McCarthy, John J., and Prince, Alan. 1995. Faithfulness and Reduplicative Identity. In *University of Massachusetts Occasional Papers 18: Papers in Optimality Theory*, eds. Jill Beckman, Laura Walsh Dickey and Suzanne Urbancyzk, 249-384. Amherst, MA: GLSA, University of Massachusetts.

Merchant, Nazarré. 2008. Discovering underlying forms: Contrast pairs and ranking. PhD. dissertation, Rutgers University, New Brunswick. ROA-964.

Merchant, Nazarré, and Tesar, Bruce. 2008. Learning underlying forms by searching restricted lexical subspaces. In *Proceedings of the Forty-First Conference of the Chicago Linguistics Society (2005), vol. II: The Panels*, 33-47. ROA-811.

Pater, Joe. 2008. Gradual learning and convergence. *Linguistic Inquiry* 39:334-345. ROA-917.

Prince, Alan. 1990. Quantitative consequences of rhythmic organization. In *CLS26-II: Papers from the Parasession on the Syllable in Phonetics and Phonology*, eds. Karen Deaton, Manuela Noske and Michael Ziolkowski, 355-398. Chicago, IL: Chicago Linguistics Society.

Prince, Alan. 2002. Entailed Ranking Arguments. Ms., Rutgers University, New Brunswick. ROA-500.

Prince, Alan, and Smolensky, Paul. 2004. *Optimality Theory: Constraint interaction in generative grammar*. Malden, MA: Blackwell.

Prince, Alan, and Tesar, Bruce. 2004. Learning phonotactic distributions. In *Constraints in Phonological Acquisition*, eds. René Kager, Joe Pater and Wim Zonneveld, 245-291. Cambridge: Cambridge University Press.

Rosenblatt, Frank. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65:386-408.

Rosenthall, Sam. 1994. Vowel/glide alternation in a theory of constraint interaction. PhD. dissertation, University of Massachusetts, Amherst.

Tesar, Bruce. 2004. Using inconsistency detection to overcome structural ambiguity. *Linguistic Inquiry* 35:219-253.

Tesar, Bruce. 2006. Learning from paradigmatic information. In *Proceedings of the 36th Meeting of the North East Linguistics Society*, eds. Christopher Davis, Amy Rose Deal and Youri Zabbal, 619-638: GLSA. ROA-795.

Tesar, Bruce. 2011. Learning phonological grammars for output-driven maps. In *Proceedings of the 39th Annual Meeting of the North East Linguistics Society*, eds.

Suzi Lima, Kevin Mullin and Brian Smith, 785-798. Amherst, MA: GLSA. ROA-1013.
Tesar, Bruce. in press. *Output-Driven Phonology*. Cambridge Studies in Linguistics. Cambridge: Cambridge University Press.
Tesar, Bruce, and Smolensky, Paul. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.
Wexler, Kenneth, and Culicover, Peter. 1980. *Formal Principles of Language Acquisition*. Cambridge, MA: MIT Press.

Department of Linguistics
18 Seminary Place
Rutgers University
New Brunswick, NJ 08901-1184

tesar@rutgers.edu