

Learning underlying forms by searching restricted lexical subspaces

Nazarré Merchant & Bruce Tesar
Rutgers University

Section 1. Introduction

Two intertwined tasks that face a learner are learning a lexicon and learning a constraint ranking. Given a set of surface forms, it is not possible in general to determine what ranking produces the given forms without knowing what the underlying forms are and similarly the underlying forms are not discernable without information about the ranking that produced the given surface forms (Albright and Hayes 2002; Hale and Reiss 1997; Tesar and Smolensky 2000; Tesar et al. 2003). Exhaustive searching of the possible underlying forms and constraint rankings is untenable. There simply are too many possible combinations of lexica and constraint rankings; the search space is too large. Because of this the learner must adopt some strategy to search the space non-exhaustively.

In this paper we propose an algorithm whereby the learner attends to contrast pairs. Contrast pairs are pairs of surface forms that contrast in some surface feature and differ in exactly one morpheme, as described in Tesar 2004. Because these pairs contrast it follows that they must differ underlyingly in some featural specification; the surface contrast could not obtain if the pair had identical underlying specifications. Furthermore, because they differ in only one morpheme the difference in underlying specification of the words must be a consequence of different underlying specifications for the differing morphemes.

The two English words ‘cats’ and ‘cads’ form a contrast pair. Here the roots ‘cat’ and ‘cad’ are in the morphological environment plural. Because the plural morpheme has one underlying specification the surface contrast must come about from differing underlying specifications of the morphemes ‘cat’ and ‘cad’.¹

$$(1) \quad kæt + s \sim kæd + z$$

In this algorithm, for each contrast pair, the learner tests all possible underlying forms for consistency. The learner then sets the underlying values of those features that have the same value for all consistent underlying forms. The learner processes contrast pairs in order by the number of undetermined features. This way the learner processes pairs with the fewest undetermined features (the ones requiring the least computational effort) first.

There are two main ideas captured in the algorithm presented here. First, relevant information about the lexicon can be extracted by only considering contrast pairs. At no time does the learner consider more than two surface forms simultaneously, nor does it

¹ Although in this contrast pair the surface forms differ only in one feature, the surface forms of a contrast pair may differ in multiple features, and such pairs can be profitably used by the algorithm described here.

consider single forms in isolation, as does the Surgery Learning Algorithm (henceforth SLA) (Tesar et al 2003). By using contrast pairs, the algorithm can be shown to generalize the capabilities of the CA procedure, being able to set all features that the CA procedure would set and others as well.

The second main idea is that processing with contrast pairs permits a meaningful reduction in the computational effort required of the learner, relative to a simultaneous evaluation of all the data. For each contrast pair, the learner is only considering combinations of values for the unset features of the morphemes in that pair. At no time does the learner consider all combinations of unset features of all morphemes simultaneously. It is in this way that the learner avoids the combinatorial nightmare of exhaustive search.

The results reported here support the claim that using contrast pairs in learning provides a good trade-off between information and computational effort. Contrast pairs provide enough information about underlying forms to allow the learner to ultimately learn the correct grammar without requiring excessive computational effort to process.

Section 2. The algorithm

Section 2.1. Learning assumptions

The learner is assumed to receive morphologically analyzed surface forms along with full knowledge of the linguistic system, including all the constraints and all the features. The learner knows the identity of the morphemes on the surface but has no prior knowledge of the underlying specifications for those morphemes.

Prior work (Tesar 2004) on contrast pairs has demonstrated valuable properties of such pairs for linguistic systems meeting certain conditions. We adopt those conditions here for working purposes. First, correspondences both input-output and surface-surface are strictly one-to-one and onto. This implies that the linguistic systems ban deletion and insertion. Second, faithfulness constraints only preserve feature identity under correspondence; all faithfulness constraints are IDENT constraints (McCarthy & Prince 1995). This precludes constraints such as output-output correspondence (Benua 1997) and anti-faithfulness (Alderete 1999). Third, all features are binary.

The task facing the learner is then to determine the underlying specifications for each morpheme and determine the ranking that will produce the given surface forms for those underlying representations. The algorithm proposed here does this by breaking the learning task into three stages, an initial (incomplete) lexical assignment, then the assignment of further underlying features on the basis of contrast pairs, and finally assignment of default values to remaining unset features and determination of the final ranking.

Section 2.2. Initial lexical assignment of consistent featural instances

In the initial lexical assignment the algorithm attends to morphemes that have featural values that never alternate. Non-alternating features are set to the featural value that is

consistently realized on the surface (Tesar et al. 2003). All other featural instances are left unset. Setting of consistently valued featural instances is guaranteed to never set an incorrect featural value. The remaining effective lexical search space consists only of alternating features.

Section 2.3 Contrast pair processing

The main body of the algorithm has three steps that are repeated until no further lexical changes occur. After the initial lexical assignment some surface forms may have all of their featural values set. Information about the target language can be gained from these forms. The first step of the main body of the algorithm builds on this fact. Surface forms that are fully specified have error-driven learning using Biased Constraint Demotion (Tesar, 1997, Prince and Tesar 2004) applied to them. This will create a set of winner-loser pairs. These winner-loser pairs represent the ranking restrictions imposed by the knowledge of forms that have their underlying specifications fully determined. The winner-loser pairs will be carried forward throughout the algorithm, restricting possible rankings.

The second step here chooses which contrast pair to consider. The algorithm chooses that pair of surface forms that has the least number of unset features (in the case of a tie an arbitrary pair from the set of pairs with the least number of unset features is chosen). Also, as this stage is repeated, the learner disregards those contrast pairs that have led to no lexicon change since the last lexicon modification. By choosing the pair with the least number of unset features the computational effort required to process the pair will be minimized.

The third step in this sequence is to test each combination of values for unset features of the selected pair. For each such combination, which we will call a *local lexicon*, the algorithm applies error-driven learning using Biased Constraint Demotion (BCD) to the two surface forms of the pair. BCD has the side effect of detecting inconsistency; learning will fail if a local lexicon is not consistent with any ranking and such an inconsistent local lexicon is not possibly correct. The learner further restricts the possible rankings by including in BCD the winner-loser pairs created in step one thus possibly rendering more local lexica inconsistent.

For each of the lexica that are consistent each unset feature value is examined. If the value of the feature is the same across consistent local lexica, then the value of that feature is permanently set in the lexicon.

After completing the third step the algorithm returns to step one (assuming either that some feature was set in the lexicon or there remains some contrast pair that the algorithm has not considered since the last lexical modification). All steps are then repeated. The first step may now include new surface forms since the last step may have set new featural values in the lexicon. Error-driven learning then may produce new winner-loser pairs (along with maintaining the previously generated winner-loser pairs) that will further restrict possible ranking for local lexica in the third stage.

Selection of the new contrast pair proceeds as in the first pass, the contrast pair with the least number of unset features is chosen. This may not be the pair that in the previous pass had the second least number of unset features for the simple fact that the lexicon may have changed and hence surface forms which contain the morpheme (or morphemes) that were modified in the last pass have fewer unset features than in the previous pass of the algorithm. Error-driven learning then proceeds with the newly chosen environment pair and the possibly newly expanded winner-loser set.

The algorithm then proceeds to loop through these three steps until no further changes have been made to the lexicon and all contrast pairs have been considered since the last lexical modification.

Section 2.4. Final lexical assignment to default values

After the main body of the algorithm, some featural instances may not have been specified. For each feature we assume there is a default value. The algorithm assigns the default value to each of these unspecified featural instances. At this point the algorithm has completely determined a lexicon and must then produce a constraint ranking. Error-driven learning using BCD is once again performed on all surface forms using now completely determined lexicon. If final lexical assignment did not assign any necessarily incorrect feature values then error-driven learning is guaranteed to produce a correct ranking. At this point the learner is finished, having produced a ranking and lexicon.

Section 2.5 Contrast pair processing in a linguistic domain

To see specifically how the algorithm processes a contrast pair it is useful to consider a specific linguistic domain. Here we consider a simple linguistic system and a single contrast pair and see how the algorithm considers all possible lexica for the pair and sets a value in the lexicon.

Consider a linguistic system with a single binary feature, stress, and the following three constraints, the first two being markedness constraints (McCarthy & Prince 1993) and the last being an IDENT constraint (McCarthy and Prince 1995).

- (2) MAINLEFT Main stress must fall on the leftmost syllable.
- (3) MAINRIGHT Main stress must fall on the rightmost syllable.
- (4) IDENT(stress) Syllables must be identical to their correspondents in stress.

The system here restricts surface forms to bi-syllabic words formed from mono-syllabic roots and suffixes. This allows for four distinct underlying forms for morphemes, the two roots **rá** and **ra**, and the two suffixes **sá** and **sa**. From these four morphemes this system can only produce two phonotactically distinct surface forms, *rása* and *rasá*.

Now, suppose the learner is presented with the contrast pair $rá_1sa_1 \sim ra_1sá_2$. In this pair both surface forms have the same root, **ra**₁, but differing suffixes, **sa**₁ and **sa**₂. The learner knows what the morphemic decomposition of the surface forms is but does not know what the underlying specifications are; that is, whether **ra**₁, **sa**₁, and **sa**₂ are

underlyingly stressed or not. There are 8 local lexica for this pair. The learner will consider each of the local lexica in turn, determining whether each is consistent.

For the first local lexicon, with all three morphemes underlyingly unstressed, the learner determines that this particular local lexicon is inconsistent. Applying BCD to these two surface forms with **ra₁**, **sa₁**, and **sa₂** all underlyingly unstressed shows this to the learner. The results of BCD are given in the Table 1. For each feature we represented the stressed value as plus, +, and the unstressed value as minus, -.

Table 1. Inconsistent lexicon of all unstressed morphemes, **ra₁** [-], **sa₁** [-], and **sa₂** [-]

		IDENT(stress)	MAINLEFT	MAINRIGHT
1	rá ₁ sa ₁ ~rasá	e	W	L
2	ra ₁ sá ₂ ~rása	e	L	W

Line 1 requires that MAINLEFT be ranked above MAINRIGHT, while line 2 requires that MAINRIGHT be ranked above MAINLEFT. Clearly, no ranking of these three constraints can produce something consistent with these two requirements.

The learner then repeats this procedure for all eight local lexica for these two surface forms. The results are given in Table 2.

Table 2. Results of BCD on all eight local lexica

Stress values			
ra ₁	sa ₁	sa ₂	Consistent
-	-	-	no
-	-	+	yes
-	+	-	no
-	+	+	no
+	-	-	no
+	-	+	yes
+	+	-	no
+	+	+	no

Only two of the eight local lexica are consistent. That is, for the two local lexica marked consistent, there is a ranking that produces the attested surface forms from this given contrast pair. These rankings may not be the same, what matters is that they exist. For the remaining six local lexica no ranking yields the attested patterns – these local lexica are not possible underlying forms for these surface forms.

After determining which local lexica yield a consistent ranking the learner attends to the values of the features for the consistent local lexica. The learner determines which featural values are the same across consistent local lexica. In this instance the featural values for stress of **sa₁** and for **sa₂** are the same across the two consistent local lexica. The stress value for **sa₁** is unstressed, [-], and the value for **sa₂** is stressed, [+]. Because

these two featural instances have the same value across consistent local lexica, the learner sets these values in the lexicon.

After processing this contrast pair, the values of \mathbf{sa}_1 and for \mathbf{sa}_2 are set and the value of \mathbf{r}_1 remains unset. The value for \mathbf{r}_1 is left unset because the featural values of stress for \mathbf{r}_1 in consistent local lexica are not the same; in the first consistent local lexicon it is unstressed, while in the second it stressed.

At this point the learner is done processing this particular contrast pair. The learner then continues processing further contrast pairs by first attending to the fully specified surface forms and then choosing another contrast pair to process.

Section 2.6. Summary of algorithm

The algorithm in its entirety including all three steps is presented here.

Stage 1: Initial lexicon construction:

Set features that never alternate to match their surface realizations.

Stage 2: Contrast pair processing:

Repeat the following steps until no further changes to the lexicon will result.

Step 1:

Apply error-driven learning on all surface forms that are fully specified.

Retain the resulting winner-loser pairs.

Step 2:

Select the contrast pair that has least number of unset features.

Step 3:

For each local lexicon for the given contrast pair, apply error-driven learning to the contrast pair along with the winner-loser pairs from step 1.

Determine which local lexica are consistent.

For each featural value in each morpheme, if it has the same specification in each of the consistent local lexica, set that value in the lexicon.

Stage 3: Final lexical assignment and ranking production:

Step 1:

Assign default values to all unspecified features.

Step 2:

Apply error-driven learning to the surface forms using the now fully specified lexicon to produce a final ranking.

Section 3. Properties of assignment

The number of local lexica for any pair is significantly less than the number of possible full lexica for all morphemes. There are two reasons for this; first a local lexicon contains fewer morphemes than the full lexicon. Second, features which have already been set either by initial lexicon construction or by previously processed contrast pairs do not contribute to the number of local lexica for a given pair. For example, in the system in section 4 below, each morpheme has two binary features, and each word consists of two morphemes, a root and a suffix. Thus there is a theoretical maximum of four distinct

roots and four distinct suffixes in this system, for a full lexicon of eight morphemes. Because each morpheme has two features the number of possible full lexica is $2^{16} = 65,536$. In a contrast pair, the two surface forms share one morpheme and differ on another yielding a combined total of three morphemes for the pair. If we assume temporarily that none of the features have been set, the number of possible local lexica for the pair is $2^6 = 64$. There are at most 48 contrast pairs for a language in this system, so the theoretical maximum number of local lexica to be evaluated is $48 * 64 = 3072$. This is significantly less than the number of possible full lexica. Furthermore, this is a gross overestimate of the number of local lexica that would be evaluated by the algorithm. Any features which do not alternate will be set by the initial lexicon construction and will not contribute combinatorially to the number of local lexica. Further, features which are set by contrast pairs which are processed early will not contribute combinatorially to the number of local lexica for later pairs that contain the same features. Finally, some pairs of surface forms may differ only in one morpheme but have identical surface realizations, and thus are not contrast pairs; these will never be processed.

Of the three stages of the algorithm, the first two, initial lexicon construction and contrast pair processing, will never incorrectly set the value of any underlying feature. The correctness of the first stage follows from the assumptions about the properties of the linguistic systems (Tesar et al. 2003).

In the second stage, a featural value for a morpheme in a contrast pair is set in the lexicon when it has a uniform value underlyingly across all consistent local lexica for that contrast pair. Crucially, one of the considered local lexica is the target local lexicon. This is because all features set to this point are guaranteed to be set correctly, and all combinations of values for the unset features are considered. The learner, of course, does not know which of the local lexica is the same as the target local lexicon, only that one of them is correct. Once the learner has evaluated all of these local lexica for consistency, it examines the feature values for only the consistent ones. The target lexicon will, necessarily, be consistent. For a featural value to be set it must have the same underlying value across all consistent local lexica and hence the same value as the value in the target local lexicon. Therefore no featural values will be set incorrectly during the second stage.

The order in which contrast pairs are evaluated can potentially affect the total amount of computational effort (which is why the algorithm processes pairs with the fewest unset features first). However, the pair evaluation order cannot affect the ultimate feature values set. Stage 2 will set the same features to the same values regardless of pair evaluation order.

It is possible that during stage 3, the final default setting of featural values and final determination of the final ranking, an incorrect value might be assigned. This will occur when a contrastive feature necessarily requiring the non-default value is not set during the first two stages. This incorrect setting would be detected while doing error-driven learning over the entire set of surface forms. Because the only possible source of

inconsistency is the final default assignments at the end of the algorithm the learner has knowledge of which featural values are possibly incorrect. We don't explore the issue further here, but in such a case other strategies could be applied to correct the values of the features set during stage 3. One possibility would be to test for consistency all combinations of values for the features not set in stages 1 and 2. Another possibility would be to use some other procedure for simultaneously evaluating lexica and rankings, such as the Surgery Learning Algorithm.

Section 4. Application of the algorithm to an enriched linguistic system

Section 4.1. Linguistic system description

Features which act in complete isolation can be handled via Contrast Analysis because when two surface forms differ with respect to that feature the only possible source of the contrast is the underlying value for that feature (Tesar 2004). The algorithm described in this paper inherits this property, being capable of setting all features that Contrast Analysis can. Features which interact can create more complex patterns. We constructed a linguistic system with two features which can possibly interact in order to evaluate the ability of the current algorithm to handle such interactions. This system extends the one described in section 2.5 by adding a vowel length feature to syllables. Three more constraints are added for a total of six. The complete constraint set is listed below.

- (5) MAINLEFT Main stress must fall on the leftmost syllable.
- (6) MAINRIGHT Main stress must fall on the rightmost syllable.
- (7) *V: No long vowel.
- (8) WSP Weight to stress principle, if long then stressed.
- (9) IDENT(stress) Syllables must be identical to their correspondents in stress.
- (10) IDENT(length) Syllables must be identical to their correspondents in length.

Of the six constraints four are markedness constraints (MAINLEFT and MAINRIGHT (McCarthy and Prince 1993), *V: (Rosenthal 1994), and WSP (Prince 1990)) and two are faithfulness constraints (IDENT(stress) and IDENT(length) (McCarthy and Prince 1995)). The words of this system are composed of a mono-syllabic root plus a mono-syllabic suffix. Each morpheme has two features, stress and length. There are four possibly distinct root morphemes, one for each combination of the two binary features; the same is true for the suffixes. These eight morphemes (four roots and four suffixes) can combine to form as many as 16 words. The number of distinct words will be fewer when two or more distinct morphemic underlying forms surface identically in all contexts. This system defines 24 distinct languages which range in size from 1 to 16 words.

The algorithm was applied to all 24 languages in this system. In all languages in the system the algorithm determined a lexicon and a ranking that correctly produced the given surface forms; that is, the algorithm learned all target languages. Furthermore, in all but 4 languages the algorithm fully determined the correct lexicon by the conclusion of the contrast pair processing stage. So all necessarily contrastive features were found and set while considering contrast pairs. In the remaining 4 languages where default

values were assigned after the contrast pair processing stage of the algorithm the featural values set were not "accidentally" set correctly; the features, although alternating, were completely predictable from the constraint ranking. Hence *any* values set for these featural instances during the final lexicon assignment stage would determine a correct lexicon for these languages.

This algorithm has stages each of which uses a different strategy for assigning underlying values. In the application of the algorithm to this linguistic system each stage accomplished a certain amount of work. Since every language had featural instances that did not alternate, the first stage set values in all of the 24 languages. In fact, 6 languages exhibited no alternations whatsoever. In these languages the initial lexical assignment completely determined the lexicon. No contrast pairs were ever considered and no final setting of default values was undertaken. The correct final ranking was determined using BCD on the forms that had their lexical values determined in the initial lexical assignment.

In 14 of the languages that exhibited alternations (of 18 that had alternations), information from the processing of contrast pairs lead to a complete specification of the lexicon. No values were set in final lexical assignment stage because the lexicon had already been fully specified. So here contrast pairs yielded enough information to set the entire lexicon, and set it correctly.

In the remaining 4 languages that exhibited alternations 2 featural instances remained unset after the processing of the contrast pairs (in each of the 4 languages). Default values were assigned and error-driven learning yielded the correct hierarchy. As described above, these featural values are completely predictable in these four languages. This means that *any* values set for these features would have yielded a correct lexicon and ranking. The bulk of the work for these languages was done by the initial lexical assignment stage and the processing of the contrast pairs.

Throughout, the learner is using contrast pairs and inconsistency detection to learn information about the lexicon.

The pairs considered contain more information than a single surface form and less than all of the forms together. By focusing on pairs of surface forms the learner significantly reduces the number of local lexica considered compared to considering all possible lexica for the target language.

By using contrast pairs the learner strikes a balance, doing less work than by exhaustively searching all possible lexica while extracting sufficient information to determine the correct grammar.

Section 5. Comparisons to other approaches

Section 5.1. Comparison to Surgery Learning Algorithm

The Surgery Learning Algorithm constructs an initial lexicon in the same manner as the algorithm presented here, by setting features that never alternate to match their surface realizations. The SLA then assigns default values to all unset featural specifications. The SLA-learner proceeds by applying error-driven learning (using BCD) to each surface form individually, maintaining a set of winner-loser pairs produced from previous applications of error-driven learning to individual surface forms. If error-driven learning reaches inconsistency, SLA selects a morpheme appearing in one of the existing winner-loser pairs that has an alternating feature that is still set to its default value. The SLA then temporarily sets that feature to its non-default value. The SLA then performs “surgery”, modifying all winner-loser pairs that contain that morpheme so they are consistent with the non-default setting of the feature. If the surgery resolves the inconsistency (if the altered set of winner-loser pairs is consistent), the non-default value for the changed feature is kept permanently. If surgery does not resolve the inconsistency, the default value is reassigned to that feature and the next alternating feature is tested in the same fashion. If no single featural modification resolves the inconsistency, then the SLA fails.

The SLA may set a feature value incorrectly. This may happen when a non-default setting of a feature solves the current set of winner-loser pairs the SLA has generated, but crucially does not satisfy requirements imposed by surface forms not yet encountered. The SLA strategy can fail in this way because for a set of winner-loser pairs it only considers one possible solution at a time. If one featural change resolves the inconsistency, it is kept. It is kept even if other featural changes would also resolve the inconsistency. The SLA does not wait for further data to determine which featural change is correct. As described in section 3 above, the algorithm presented in this paper will not set an incorrect value while processing contrast pairs. It only commits to values for underlying features when they are guaranteed to be correct.

Note that it could be computationally expensive for the SLA to even test all possible combinations of values for the alternating features in the winner-loser pairs whenever inconsistency is detected. This is because the winner-loser pairs are constructed from all of the data processed to that point by the learner. The algorithm proposed here avoids this problem by processing only a single contrast pair at a time. It only considers combinations of unset features appearing in any single contrast pair.

Section 5.2. Comparison to Contrast Analysis

The Contrast Analysis learning procedure processes the same contrast pairs as does this algorithm. The CA procedure, for a given contrast pair, determines if there is a feature on which the contrasting morphemes must differ underlyingly in order to account for the contrast in the surface forms. If there is only one feature that could possibly account for the surface contrast, then the CA procedure sets (correctly) the value of that feature underlyingly; otherwise the CA procedure does not set any feature values on the basis of that contrast pair.

The criteria that the CA procedure uses for determining if a feature could account for a surface contrast are defined solely with respect to (a) the surface features on which the two surface forms of the contrast pair differ, and (b) underlying feature values that have already been set. No reference is made to information the learner may possess about the constraint ranking, or even to the identity of the constraints themselves. This makes the criteria very weak. The algorithm, when confronted with surface forms that differ on several values, may not be able to identify any one of those features as necessarily accounting for the contrast, and thus will fail to set any of them, when in fact only one of the features could possibly account for the surface contrast via one of the possible rankings.

The algorithm presented here will set all of the features that CA will set, and goes beyond it to set even more features by making use of information about the constraints and the ranking. Consider the language presented in table 3.

Table 3. The surface forms for a language in the stress-length system

ra ₁	ra ₂	
rása	rá:sa	sa ₁
rasá:	rása	sa ₂

In this language length is contrastive while stress is not. Thus there are only two roots and two suffixes with distinguishable phonological behavior. The constraint WSP is undominated in this language so long vowels only appear on the surface in stressed syllables. The default stress is on the initial syllable.

Initial lexicon construction yields the following lexicon.

- (11) The results of initial lexicon construction [+/- stress, +/- length]
 ra₁ [?, -] ra₂ [+ , +] sa₁ [-, -] sa₂ [?, ?]

The learner will attempt to set the features of sa₂ by constructing a contrast pair in which sa₂ contrasts with sa₁. In the environment of root ra₂, both suffixes surface identically, so a valid contrast pair cannot be formed. In the environment of ra₁, the suffixes surface distinctly so the learner constructs the contrast pair rá₁sa₁ ~ ra₁sá:2. In this pair, the suffixes differ on the surface in both stress and length. Furthermore, sa₂ is currently unset for both features and thus could possibly differ from sa₁ in either feature value. Thus CA is unable to set either feature; it does not know which feature is responsible for the surface contrast.

The algorithm presented here will process the very same contrast pair by constructing all eight local lexica and testing for consistency. The results of this are shown in table 4.

Table 4. Results of error-driven learning on all eight local lexica

Feature values			
ra ₁ stress	sa ₂ stress	sa ₂ length	Consistent
–	–	–	no
–	–	+	yes
–	+	–	no
–	+	+	yes
+	–	–	no
+	–	+	yes
+	+	–	no
+	+	+	yes

The algorithm observes that the sa₂ length feature is set to +long in all consistent local lexica, and thus can set that feature. It is able to succeed here where CA failed because error-driven learning makes crucial reference to the constraints. In this case, it is not possible for a morpheme to surface as long in any environment if it is underlyingly short; given the constraints, a long vowel can only appear on the surface as a consequence of faithful preservation of an underlying long vowel. CA does not use this kind of information.

Section 6. Discussion

Section 6.1. Issues

The algorithm focuses its learning efforts on extracting information from contrast pairs. These pairs serve as the primary means of determining the lexicon of the target language. A key question is, what properties a linguistic system must possess to ensure that contrast pairs are sufficient to determine the lexicon? In the stress length system of section 4, contrast pairs are definitely adequate; the information is sufficient to set all contrastive features. What properties of a system are necessary for success with contrast pairs is still an open question.

This algorithm will fail when a feature must be set to a non-default value but is never set by contrast pair processing or initial lexicon construction. One way this could happen is if in morpheme *m*, for any contrast pair *m* is in, two possible local lexica with differing values for the feature in question do not lead to inconsistency. Here the two different values of the key feature for *m* can be consistent for each environment in which *m* occurs, but crucially the conditions allowing the default feature value (the values for other features and the constraint ranking) are not the same across the different environments. Clearly if all surface forms were considered simultaneously, then only one value of the feature would be shown to be consistent, but no one contrast pair contains all of the information necessary to set the value of the feature.

Another type of system on which the algorithm can fail is one in which the surface forms for a language are consistent with more than one distinct grammar, specifically, distinct grammars with distinct settings of contrastive underlying features. In this case, contrast pair processing will not set these features because consistent grammars exist for both

values of the features. This is an inherent problem for any procedure that sets values only when certain that the values must be correct.

Section 6.2. Further directions

Given the concerns raised in the previous section about the amount of information available in any single contrast pair, it is natural to investigate ways of incorporating further information into the processing of contrast pairs that are not prohibitively computationally costly.

The problem with adding additional surface forms to a contrast pair is that in general they will involve additional morphemes with unset features. This will increase exponentially the number of local lexica. However, no such increase will occur if the additional morphemes are already fully specified underlyingly (all of their features are set). Thus one could add to the contrast pair surface forms combining morphemes of the contrast pair with other fully specified morphemes. Processing would then apply error-driven learning to all of these surface forms (the forms of the contrast pair plus the added ones). The additional surface forms could place further restrictions on both the ranking and the underlying feature values, increasing the number of local lexica that are inconsistent, and thus increasing the likelihood of setting a feature.

Another means of reducing the number of consistent local lexica for a given contrast pair lies with the phonotactics of the language. Phonotactic learning has been proposed as an early stage of learning in which the learner makes observations about the phonotactics of the language without yet being aware of morpheme identity (Hayes 2004, Prince and Tesar 2004). Phonotactic learning algorithms can avoid the issue of underlying forms for morphemes by assuming an underlying form for each entire word that is identical to the surface realization of the word. The learner applies error-driven learning to forms mapping each surface form to itself and accumulates a set of phonotactic winner-loser pairs. It applies a procedure like BCD to construct a constraint ranking that enforces the phonotactics of the surface forms. In general this is not sufficient to determine the complete ranking for a language, but it is capable of gaining partial information about the ranking. Tesar and Prince (to appear) have argued that phonotactic information, expressed in the form of phonotactic winner-loser pairs, can be used to set the underlying values of at least some features. Given the apparent usefulness of this information, the algorithm described in this paper might benefit from including the phonotactic winner-loser pairs in the processing of contrast pairs.

One property of error-driven learning is that it can fail to extract all of the ranking information from a given surface form. When this happens the learner fails to generate certain losing candidates that would form winner-loser pairs that would provide additional information about the ranking. The contenders algorithm (Riggle 2004) generates an exhaustively informative set of losers for a given surface form. One might compensate for the occasional short-comings of error-driven learning by using the contenders algorithm either to replace or to supplement error-driven learning for the purpose of constructing winner-loser pairs.

Section 7. Conclusion

The results presented here support the claim that the underlying values of contrastive features can be set on the basis of contrast pair processing. In the stress-length system discussed in section 4, every contrastive feature not already set by initial lexicon construction was set on the basis of some particular contrast pair; that single pair contained sufficient information to force the correct value of the feature. The contrast pairs, while providing sufficient information to set contrastive features, are individually dependent on the underlying forms of only the morphemes of that pair, not all of the underlying forms of the language. Thus a contrast pair can be processed while only considering different values for the unset features of the morphemes of that pair. This involves far less computational effort than the consideration of all combinations of all unset features of the language. In this way, the algorithm presented here achieves a beneficial balance between information and computational effort. A contrast pair provides enough information to be a useful unit for learning while being small enough that it can be processed efficiently.

Acknowledgements

The authors thank Paul de Lacy, Alan Prince, Jason Riggle, the Rutgers Optimality Reading Group, and the participants of the 2005 Hopkins - UMass - Rutgers Joint Class Meeting for helpful comments and discussion. This material is based upon work supported by the National Science Foundation under Grant No. BCS-0083101. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Any errors are the responsibility of the authors.

References

- Albright, Adam, and Hayes, Bruce. 2002. Modeling English past tense intuitions with minimal generalization. In *Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, ed. Michael Maxwell, 58-69. Association for Computational Linguistics.
- Alderete, John. (1999). Morphologically Governed Accent in Optimality Theory. PhD. Dissertation, University of Massachusetts, Amherst.
- Benua, Laura. (1997). Transderivational Identity: Phonological Relations Between Words. PhD dissertation. University of Massachusetts, Amherst.
- Brasoveanu, Adrian. 2004. Stress/Length Interaction: Alternations In A System With WSP. Ms. Rutgers University.
- Hale, Mark, and Reiss, Charles. 1997. Grammar Optimization: The simultaneous acquisition of constraint ranking and a lexicon. Ms., Concordia University, Montreal. ROA-231.
- Hayes, Bruce. 2004. Phonological acquisition in Optimality Theory: The early stages. In *Constraints in Phonological Acquisition*, ed. by René Kager, Joe Pater, and Wim Zonneveld, 158-203. Cambridge: Cambridge University Press.
- McCarthy, John and Prince, Alan. 1993. Generalized alignment. In *Yearbook of Morphology*, eds. Geert Booij and Jaap Van Marle, 79-154. Dordrecht: Kluwer.

- McCarthy, John and Alan Prince. (1995) Faithfulness and Reduplicative Identity. In *University of Massachusetts Occasional Papers 18: Papers in Optimality Theory*, ed. By Jill Beckman, Laura Walsh Dickey, and Suzanne Urbanczyk, 249-384. Amherst, MA: GLSA, University of Massachusetts.
- Prince, Alan. (1990) Quantitative consequences of rhythmic organization. *Chicago Linguistic Society*, 26.2, 355-98.
- Prince, Alan, and Tesar, Bruce. 2004. Learning phonotactic distributions. In *Constraints in Phonological Acquisition*, ed. by René Kager, Joe Pater, and Wim Zonneveld, 245-291. Cambridge: Cambridge University Press.
- Riggle, Jason. 2004. Generation, Recognition, and Learning in Finite State Optimality Theory. PhD dissertation UCLA.
- Rosenthal, Sam. 1994. Vowel/glide alternation in a theory of constraint interaction. PhD. dissertation, University of Massachusetts, Amherst.
- Tesar, Bruce. 1997. Using the mutual inconsistency of structural descriptions to overcome ambiguity in language learning. In *Proceedings of the North East Linguistic Society 28*, ed. by Pius N. Tamanji and Kiyomi Kusumoto, 469-483. Amherst, MA: GLSA, University of Massachusetts.
- Tesar, Bruce, and Smolensky, Paul. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.
- Tesar, Bruce, Alderete, John, Horwood, Graham, Merchant, Nazarré, Nishitani, Koichi, and Prince, Alan. 2003. Surgery in language learning. In *Proceedings of the Twenty-Second West Coast Conference on Formal Linguistics*, ed. by G. Garding and M. Tsujimura, 477-490. Somerville, MA: Cascadilla Press. ROA-619.
- Tesar, Bruce. 2004. Contrast analysis in phonological learning. Ms., Linguistics Dept., Rutgers University. ROA-695.
- Tesar, Bruce and Prince, Alan. To appear. Using phonotactics to learn phonological alternations. In *Proceedings of the 39th Conference of the Chicago Linguistics Society, Vol. II: The Panels*, Eds. Johnathon E. Cihlar, Amy L. Franklin, David W. Kaiser, and Irene Kimbara. ROA-620.