A Limited Non-Deterministic Parameter-Setting Model

Eric H. Nyberg 3rd
Carnegie Mellon University

A LIMITED NON-DETERMINISTIC PARAMETER-SETTING MODEL

Eric H. Nyberg, 3rd

Carnegie Mellon University

1. Introduction

In Chomsky's Principles and Parameters model, Universal Grammar is represented as a set of principles, or constraints on the form of possible grammars (Chomsky 1981, 1985). Cross-linguistic variation is captured by parameterized differences in the definitions of principles. For example, the binding domain for anaphoric referring expressions seems to vary in certain ways across languages, and has been analyzed as a type of parametric variation (Borer and Wexler, 1987). Other work on the parameterization of Universal Grammar includes various proposals concerning word order (Koopman, 1983; Travis, 1984), null subjects (Hyams, 1986; Jaeggli and Safir, 1989), second language acquisition (Flynn, 1987) and metrical phonology (Halle and Vergnaud, 1987; Dresher and Kaye, 1990).

If cross-linguistic variation is captured by a set of parameter values for each principle which can vary across languages, then the goal of the language learner must be to attain the correct adult settings for each of the parameters of Universal Grammar in response to early linguistic experience. A particular model of the acquisition process shall be judged not only by what grammar it attains when presented with certain data, but also by how well its process of acquisition reflects empirical facts about limitations on child memory, attention, etc. (Pinker, 1979), as well as the observed stages of child acquisition (Brown, 1973).

As an illustration of how parameters can capture cross-linguistic variation, consider just three of the parameters in the model proposed by Dresher and Kaye (1990):

- *Boundedness of Constituents (P_1).* If P_1 is 0, then constituents (feet) are bounded (binary); if P_1 is 1, then constituents are unbounded in size;

- *Direction of Constituent Construction (P_2).* If P_2 is 0, then metrical feet are constructed from left to right; if P_2 is 1, feet are constructed from right to left;
- *Headedness of Constituents (P_3).* If P_3 is 0, then constituents (feet) are left-headed; if P_3 is 1, they are right-headed.

These parameters work together to produce the 8 basic 5-syllable word types shown in Figure 1. Each entry in the table shows the metrical constituent structure produced for a 5-syllable word given each of the possible combined settings of the three parameters[1]:

L	P_0	P_1	P_2	Metrical Feet
1.	0	0	0	((A* B) (C* D) (E*))
2.	1	0	0	((A* B C D E))
3.	0	0	1	((A B*) (C D*) (E*))
4.	1	0	1	((A B C D E*))
5.	0	1	0	((A*) (B* C) (D* E))
6.	1	1	0	((A* B C D E))
7.	0	1	1	((A*) (B C*) (D E*))
8.	1	1	1	((A B C D E*))

Figure 1: **Examples of Metrical Constituent Structure**

This example illustrates two important points about parameterized models. First, note that changing the value of a single parameter can change the nature of the resulting language in a profound way. For example, the language in row 7 differs from the language in row 8 by the value of a single parameter, but the two languages have dramatically different stress patterns: language 7 has a right-headed, binary stress pattern, while language 8 has an unbounded, right-headed stress pattern. The second thing to note is that two different sets of parameters values may produce the same stress pattern for a single word. For example, both language 2 and language 6 produce the same stress pattern for the example word.

The implication for parameter setting learning models is that it is difficult to formulate simple patterns that determine the language being learned on the basis of a few example words or sentences. It is necessary for the learner to consider the overall effects of all the parameters and the stress patterns of many words of different lengths to differentiate between sets of parameter values. For this reason, it is difficult to formulate a coherent and unambiguous set of deterministic learning cues without imposing strict constraints on the developmental course of learning and the type of data available, which can lead to incorrect predictions about the nature of child language learning. For example, the deterministic learner YOUPIE developed by Dresher and Kaye can successfully learn parameters in metrical phonology, but must set the parameters in a predetermined, rigid order that is based on the learning cues themselves. YOUPIE also makes cross-word comparisons over the entire set of data, which requires significant amounts of memory and processing which may not be available to the child (Dresher and Kaye, 1990; Pinker, 1979).

In this paper, I will focus on two particular problems with deterministic, error-driven learning algorithms, and present a limited non-deterministic parameter-setting model that addresses these

[1] For the sake of illustration, I abstract away from extrametricality, quantity-sensitivity, etc. in this example. The head of each foot (which is marked with secondary stress) is indicated by an asterisk.

problems in a system for learning stress assignment parameters in metrical phonology.

2. Problems with Deterministic Error-Driven Learning

Most of the learning algorithms that have been proposed as models of language learning are *deterministic, error-driven* systems. For example, in the early model proposed by Wexler and Culicover (1980), the learner maintained a single hypothesis, consisting of a set of transformation rules, that changed only when the current example sentence (surface structure) could not be derived from the example base structure using the existing set of transformations. In Berwick's model (1985), the learner maintained a single hypothesis, consisting of a set of phrase-structure rules, which was modified only when the current example sentence could not be parsed with the current grammar. In the YOUPIE model, the learner maintains a single hypothesis, and sets its parameters one by one as the learner's cues either match or fail to be matched by the words in the input sample. In general, deterministic error-driven parameter learners are characterized by the following attributes[2]:

- The learner maintains a single active hypothesis h, initially the least-marked hypothesis (all parameters are set to 0);
- If the learner encounters an example $s \notin L(h)$, then it changes the value of a single parameter from 0 to 1;
- The learner cannot set a parameter from 1 to 0 (backtracking is not allowed);
- Each hypothesis selected by the learner obeys the Subset Principle[3].

In the remainder of this section, I shall focus on two characteristics of this type of model that cause undesirable behavior and can lead to unrecoverable errors:

- *Parameter Flipping.* Deterministic models that "flip" a parameter when a single example $s \notin L(h)$ is encountered in the input sample are sensitive to the presence of ungrammatical examples in the data. As a result, they can fluctuate between incorrect hypotheses without converging to the correct grammar.
- *Single Hypothesis.* Strictly deterministic systems maintain only a single hypothesis during learning, and do not allow backtracking (resetting a parameter to 0 once it has been set to 1). Since learning situations arise where the learner must choose between more than one possible hypothesis based on local evidence, changing the wrong parameter can lead to an unrecoverable error when backtracking is not allowed.

2.1 Parameter Flipping

Consider a learning algorithm that flips a parameter when some example sentence $s \notin L(h)$ is seen in the input. Such an algorithm will always choose a new hypothesis when it encounters a sentence outside its hypothesized language. If we assume that the input data contain only grammatical example sentences from the target language, then the learner can safely assume that the presence of some $s \notin L(h)$ in the input means that the current hypothesis h is incorrect and that $L(h)$ is not the target language. In reality, the problem faced by child language learners is not so ideal. The input data processed by children contain at least occasional performance errors, restarts, ungrammatical sentences, etc. It is unlikely that an algorithm that always flips a parameter in the presence of some $s \notin L(h)$ will be able to converge to the correct hypothesis.

[2]In this paper, I will use s to denote an input word or sentence presented to the learner, h to denote a hypothesis held by the learner (a set of parameter values), and L to denote the language generated by a particular hypothesis.

[3]Note that this is vacuous, if the the least-marked value of a subset parameter is always the subset value.

Input Example	Hypothesis
⋮	⋮
$s_m \in L_{target}$	L_{target}
$s_{m+1} \in L_{target}$	L_{target}
⋮	⋮
$s_n \notin L_{target}$	L_j
$s_{n+1} \in L_{target}$	L_k
$s_{n+2} \in L_{target}$	⋮
⋮	⋮

Figure 2: **Fluctuation Between Incorrect Hypotheses**

An example learning scenario is illustrated in Figure 2. At some time m, the learner encounters a piece of data s_m that is in the target language, and hypothesizes the target language, L_{target}. At this point it will maintain the correct hypothesis for any number of subsequent input examples that are in L_{target}. However, suppose that at time n the learner encounters some ungrammatical example sentence S_n that is not in the target language L_{target}. Since the learning algorithm flips a parameter whenever it sees some $s \notin L(h)$, it must change its hypothesis away from the correct hypothesis. Suppose that the learner chooses some hypothesis L_j. Since backtracking isn't allowed, the learner will never regain the correct hypothesis once it has abandoned it. Even if backtracking were allowed, the learner would fluctuate between the correct hypothesis and some other hypotheses, since every $s \notin L_{target}$ would cause it to select some $L' \neq L_{target}$.

2.2 Strict Determinism

In a strictly deterministic learning system, the learner may hold only a single hypothesis at any one time (corresponding to a single set of parameter values), and it may not backtrack by unsetting a parameter that it has already set. In such a system, changing the value of the wrong parameter can lead to an unrecoverable error in certain situations. As a result, the learner converges to an incorrect hypothesis.

The example learning scenario illustrated in Figure 3 contains just two parameters, and therefore the learner must choose between 4 possible hypotheses[4]. Let us assume that the first parameter is a subset parameter; as a result, $L(H_1) \subset L(H_3)$ and $L(H_2) \subset L(H_4)$. Suppose that the learner starts with H_1, the least-marked hypothesis (both parameters are set to 0), and that $L(H_2)$ is the target language. Since $L(H_1)$ is not the target language, the learner will soon encounter some $s \notin L(H_1)$. It will then flip one of its parameters. The question is, which one? There are two possibilities; setting the first one will yield $\langle 1, 0 \rangle$, or $L(H_2)$, and setting the second one will yield $\langle 0, 1 \rangle$, or $L(H_3)$. A strictly deterministic learner must pick just one of the two possibilities, since it can hold only a single hypothesis at one time. Suppose that the learner picks not H_2, but H_3. It will soon encounter example sentences not in $L(H_3)$. Since backtracking is not allowed, the learner must then set the second parameter, yielding $\langle 1, 1 \rangle$, or $L(H_4)$. Note that since $L(H_2) \subset L(H_4)$, there is no way for the learner to reach the correct hypothesis without backtracking. However, even if backtracking were allowed, the learner would never encounter data that would prompt it to select

[4]This example is drawn from Clark's discussion of the Causality Problem (Clark, 1988).

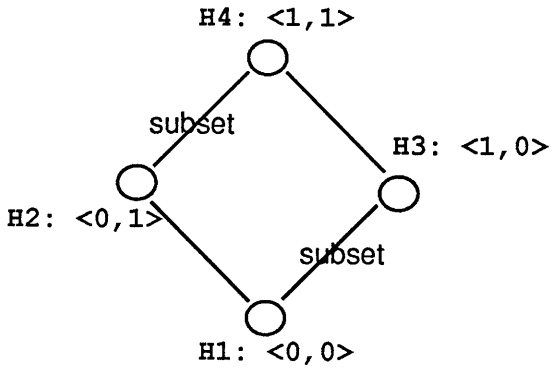

Figure 3: **A Simple Hypothesis Ordering Problem**

$L(H_2)$ once $L(H_4)$ is selected. Since all subsequent input examples $s \in L(H_2)$ are also in $L(H_4)$, the superset language, the learner would (falsely) assume that it had converged to the correct language.

A successful learning algorithm must avoid "traps" like these and converge to the correct target language, even when there are occasional ungrammatical examples in the input or when the learner must choose between more than one hypothesis at a given time. It seems that a learner which picks the smallest language that "fits" the data (i.e., minimizes $s \notin L(H)$) would be able to avoid both of these problems. Unfortunately, a learning model that "remembers" all the example sentences is cognitively implausible, and therefore not of interest to us (Pinker, 1979). However, it is of interest for us to explore learning algorithms that show similar behavior, in particular, the ability to "ignore" infrequent ungrammatical examples that might otherwise deceive the learner and cause it to abandon the correct hypothesis.

The model described below uses a method of weighing evidence for or against particular hypotheses. Without "remembering" all the input data, it can avoid the parameter flipping problem. In addition, the model makes use of limited non-determinism in order to avoid hypothesis ordering problems like the one previously discussed.

3. A Uniform Hypothesis Weighting Mechanism

In this model, each hypothesis held by the learner has two components: a string of 0's and 1's indicating a particular set of parameter values, and a *weight*, a numerical value which indicates the level of confidence in that particular hypothesis. Intuitively, a higher weight indicates that there is more evidence for a particular hypothesis than one with a lower weight. This notion is made concrete by the following definitions:

1. The set $DATA_t$ is the set of example sentences encountered by the learner up to time t;
2. The set $POS_t = s \in L(h), s \in DATA_t$;
3. The set $NEG_t = s \notin L(h), s \in DATA_t$;
4. The value of $NPE_t(h) = |POS_t(h)| - |NEG_t(h)|$.

In other words, the *net positive evidence* (NPE) in support of a particular hypothesis can be calculated by counting the number of examples sentences in $L(h)$ and subtracting the number of example

sentences not in $L(h)$. For example, if at time t, $DATA_t = \{s_1, s_2, s_3\}$, $s_1, s_2 \in L(h_i)$, $s_3 \notin L(h_i)$, then $NPE_t(h_i) = |\{s_1, s_2\}| - |\{s_3\}| = 2 - 1 = 1$.

The weight associated with each active hypothesis at time t is a function of $NPE_t(h)$. The particular family of functions used for weighting in this model is the set of sigmoid functions, $y = SIG(x) = 2/(1 + e^{-kx}) - 1$, where k is a damping factor that controls the slope of the curve (see Figure 4)[5].

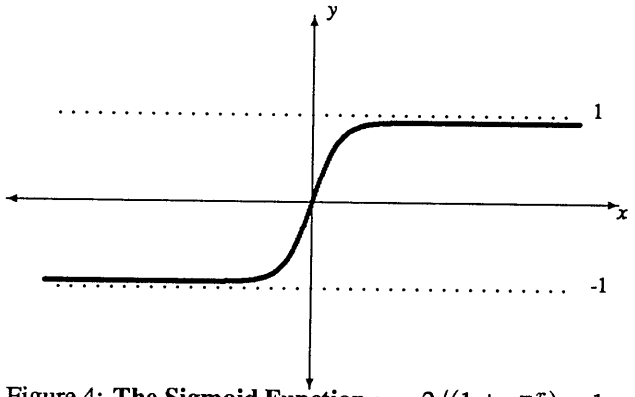

Figure 4: **The Sigmoid Function** $y = 2/(1 + e^{-x}) - 1$.

This weighting function has two desirable characteristics:

- The curve approaches -1 as NPE gets more and more negative, and +1 as NPE gets more and more positive; hence weights of -1 and +1 represent our intuitive notions of complete lack of confidence in a hypothesis vs. complete confidence in a hypothesis;
- Because the slope of the curve becomes more and more shallow as it approaches its asymptotes, fluctuations in $NPE(h)$ near the asymptotes will have little effect on the weight of h.

The learner checks each active hypothesis h against each new piece of data s; if $s \in L(h)$, then h receives positive weight, otherwise h receives negative weight as determined by the sigmoid function. When many examples are encountered that are not in $L(h)$, the weight of h will get closer and closer to -1. If the weight of a hypothesis gets close enough to -1, it will be removed from the list of active hypotheses. If the weight of a hypothesis gets close enough to +1, the learner has converged to that hypothesis. To formalize the notion of "close enough," I assume the existence of some threshold ϵ, such that hypotheses with weights less than $(\epsilon - 1)$ are removed from the active list and hypotheses with weights higher than $(1 - \epsilon)$ are selected by the learner as final hypotheses. For example, if $\epsilon = .1$, then hypotheses with weights less than -.9 will be removed from the active list, and hypotheses with weights greater than .9 will be selected as final.

Consider the learning scenario illustrated in Figure 5. At some time m, the learner encounters $s_m \in L_{target}$, and begins to accumulate positive weight for h_{target}. After many more examples are processed, the weight of h_{target} will approach +1, as observed here at time n. Suppose that at time $n + 1$ the learner encounters $s_{n+1} \notin L_{target}$. Using the weighting mechanism sketched

[5]There are certainly other types of weighting functions that one might consider; this is only one such function that has the desired characteristics.

Input Example	Hypothesis
⋮	⋮
$s_m \in L_{target}$	$W(h_{target}) = .001$
$s_{m+1} \in L_{target}$	$W(h_{target}) = .025$
⋮	⋮
$s_n \in L_{target}$	$W(h_{target}) = .898$
$s_{n+1} \notin L_{target}$	$W(h_{target}) = .887$
$s_{n+2} \in L_{target}$	$W(H_{target}) = .898$
⋮	⋮

Figure 5: **Uniform Weighting and Ungrammatical Examples**

above, the learner will subtract only a small amount of weight from h_{target}, since on the whole the data support h and it has accumulated much positive weight. The learner does not therefore flip a parameter (change its hypothesis) on the basis of a single example; rather, it is the cumulative effect of larger amounts of data that tend to confirm or disconfirm particular hypotheses over time, making the learner much more resilient to the presence of infrequent ungrammatical examples.

The uniform weighting strategy was tested on a small learning problem containing two word order parameters and four hypotheses corresponding to the basic word orders (SOV, OVS, SVO, VOS) (Nyberg, 1987, 1989). An experiment was conducted in which 200 grammatical examples of SVO sentences were presented to the learner. The weight associated with the correct hypothesis is shown in the graph in Figure 6.

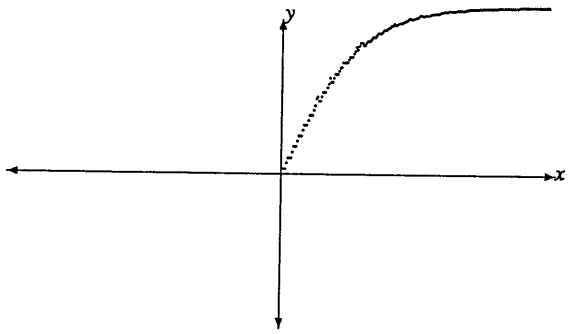

Figure 6: **Weight Assigned to Word Order SVO Over 200 Examples**

As predicted, the learner's confidence in that hypothesis grows steadily and approaches 1 (complete belief). Another experiment was conducted, in which another 200 examples were presented to the learner, but this time with a 10% error rate (roughly 1 out of every 10 examples was ungrammatical, i.e., it could not be parsed successfully into an SVO sentence). The weight associated with the SVO hypothesis is shown in Figure 7. The learner was still able to converge to the correct hypothesis, but the level of weight assigned to the SVO hypothesis after 200 examples is somewhat less than when

learning without errors, and the learner's belief in the SVO hypothesis can be seen to fluctuate near the origin more in Figure 7 than in Figure 6. Hence the model predicts that the learner should still be able to set parameters correctly when ungrammatical examples are present, but that it may require more data before converging to the correct hypothesis.

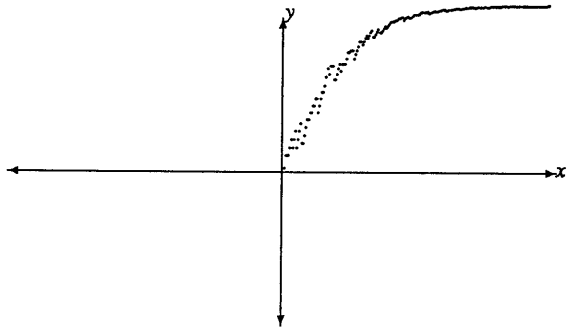

Figure 7: **Weight Assigned to Word Order SVO Over 200 Examples, 10% Error Rate**

4. Limited Non-Determinism

In order to avoid the hypothesis ordering problem shown earlier in FIgure 3, the model relaxes the assumption of strictly deterministic learning. In particular, the learner may hold more than one local hypothesis while resetting parameters. In a deterministic (single-hypothesis) learner, changing the current hypothesis can be achieved by "flipping" a value or values in the string of 0's and 1's encoding the current hypothesis. For example, 0101 can be derived from 0000 by flipping the second and fourth parameter values. In the type of non-deterministic learner described here, more than one hypothesis may be active at a given time. Adding new hypotheses can be achieved by copying one of the current hypotheses, and then flipping some of its values. For example, $\{0000, 0101\}$ can be derived from $\{0000\}$ by making a copy of 0000, flipping its second and fourth parameter values, and then storing it with the original hypothesis (thus indicating that both are active).

It should be clear from the foregoing discussion that there are no *a priori* limitations on how a learner may select new hypotheses or add to the current set of active hypotheses. It is the case, however, that child language learners do not fluctuate wildly in their choice of grammar (though they may experiment with certain ungrammatical ways of building sentences). In fact, their grammatical development seems to move smoothly through several well-defined stages. For this reason, I will adopt the following constraint on hypothesis selection:

> *The Single-Value Constraint.* At any time t, if the learner holds some hypothesis h, then it may activate only hypotheses that differ from h by the value of a single parameter (Clark, 1990).

Intuitively, the Single-Value Constraint limits the learner to only those hypotheses that can be derived by flipping a single parameter. This has two desirable effects:

- The learner cannot jump between hypotheses that are completely unrelated;

- To derive a particular parameter setting, the learner must pass through a number of intermediate stages.

The Single-Value Constraint limits the search performed by the learner in a way that makes it more plausible as a cognitive model of acquisition than an unconstrained non-deterministic learner. I will adopt the notion of *1-adjacency* to describe two hypotheses that meet the Single Value Constraint; e.g., h_i is 1-adjacent to h_j iff h_i is identical to h_j except for the value of a single parameter.

I will assume that the learner begins with the least-marked hypothesis, a string of parameter values containing only 0's. For example, in a system with 4 parameters, the initial hypothesis would be 0000. This hypothesis is the only member of the initial set of active hypotheses maintained by the learner. The goal of the learner is to search the set of possible hypotheses in order to locate the correct hypothesis. This is accomplished by weighting hypotheses as described in the previous section, checking each active hypothesis against the current input datum and weighting it accordingly. This process continues until either 1) one of the hypotheses attains a weight $> (1 - \epsilon)$, in which case the learner has converged to that hypothesis as its final grammar, or 2) there are no more active hypotheses (the weights of all active hypotheses are $< (\epsilon - 1)$). In the former case, the learner stops processing and returns the hypothesis it has selected. In the latter case, the learner activates new hypotheses according to the Single-Value Constraint. When the last active hypothesis is removed from the active list, all hypotheses that are 1-adjacent to it are activated and placed into the active list, and processing continues. For example, if 0000 is not the correct hypothesis, sooner or later the learner will accumulate enough evidence against it, and its weight will dip below $(\epsilon - 1)$, causing it to be removed from the active list. Since it is the only active hypothesis initially, this would in turn cause all 1-adjacent hypotheses to be activated, e.g., 1000, 0100, 0010, and 0001 (cf. Figure 8). The learner continues to activate and prune hypotheses in this fashion until it converges to the correct hypothesis.

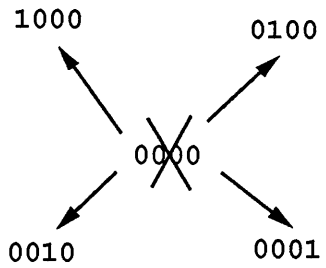

Figure 8: **Limited Non-Deterministic Search**

Although the learning algorithm is non-deterministic, the activation of new hypotheses is tightly constrained. The learner activates new hypotheses only when all of its current hypotheses are pruned. When new hypotheses are activated, the learner obeys the Single-Value Constraint, activating only those hypotheses that are 1-adjacent to the last active hypothesis. This implies that there are at most n hypotheses active at any given time, where n is the number of parameters to be learned.

Returning to the learning problem presented in Figure 3, we can see how limited non-determinism can solve the hypothesis ordering problem. When the learner decides that H_1 cannot account for the input data, it need not limit itself to picking either H_2 or H_3; since both are local to H_0, it can activate both and assume that the evidence provided by subsequent input examples will

serve to discriminate between them.

5. Learning Parameters in Metrical Phonology

In order to test the learning algorithm described above, I have replicated the parameter model presented in (Dresher and Kaye, 1990) by creating a computer program that builds metrical constituents and assigns word stress based on the current set of parameter values. This program also incorporates both the uniform weighting mechanism and limited non-determinism I have described. The set of parameters learned by the model is shown in Figure 9.

Parameter	Principle
P0	The word-tree is strong on the [Left/Right]
P1	Feet are [Binary/Unbounded]
P2	Feet are built from the [Left/Right]
P3	Feet are strong on the [Left/Right]
P4	Feet are quantity sensitive (QS) [No/Yes]
P5	Feet are QS to the [Rime/Nucleus]
P6	A strong branch of a foot must itself branch [No/Yes]
P7	There is an extrametrical syllable [No/Yes]
P8	It is extrametrical on the [Left/Right]
P9	A weak foot is defooted in clash [No/Yes]
P10	Feet are noniterative [No/Yes]

Figure 9: **Metrical Parameters from (Dresher & Kaye, 1990)**

The example shown in Figure 10 serves to illustrate how the metrical processor assigns stress to an input example based on the current set of parameter values. The example word *yangarmata* is from Maranungku, which has a left-headed, binary stress pattern. First the processor groups the syllables in the word into binary feet, as shown in the first layer or processing; then it assigns word-level stress, as shown in the second layer. Here stress devolves on the head of the left-most foot.

The stress processor is integrated with the learning algorithm in a learning system for metrical parameters, as shown in Figure 11. The stress processor passes two pieces of information to the learning algorithm: the observed stress pattern associated with the input word, and the stress pattern output by the metrical processor for the same syllables (like YOUPIE, the metrical processor strips off the stress markings from the input word and stores them for comparison with the pattern assigned by the current parameter settings). In addition, the learning algorithm also has access to the current set of parameter values, which it can modify during learning.

It is important to note that the amount of processing performed on the two stress patterns (input and output) is minimal: they are simply checked to see if they match. If not, the learner will subtract weight from the current hypothesis. If the patterns do match, then the current hypothesis successfully accounts for the current piece of data, and receives additional weight.

For the purposes of testing the learner, I have used words of length 3, 4 and 5 syllables. A set of data for the learner contains equal numbers of words of length 3, 4, and 5 for each language tested, with the appropriate stress pattern for that language assigned to the each word. In the test

```
Input segments: #([YAN]<0> [GAR]<0> [MA]<0> [TA]<0>)

Metrical constituents (LEFT-headed, BINARY feet built from the LEFT):

   [  1        0      ][  1        0      ]
     / \      / \        / \      / \
    Y   *    G   *      M   *    T   *
       / \      / \        / \      / \
      A   N    A   R      A        A

Word tree (strong on the LEFT):

   [  2 +      0      ][  1        0      ]
     / \      / \        / \      / \
    Y   *    G   *      M   *    T   *
       / \      / \        / \      / \
      A   N    A   R      A        A

((Y A N 2) (G A R 0) (M A 1) (T A 0))
```

Figure 10: **Example Output of Metrical Processor**

samples used so far, 3, 4 and 5-syllable words appear with equal frequency. Since the Dresher and Kaye model being replicated can recognize four different basic syllable types, there are $4^3 = 64$ 3-syllable words, $4^4 = 256$ 4-syllable words, and $4^5 = 1024$ 5-syllable words. In order to create a data file with equal numbers of words of each length, the 3-syllable words were duplicated 16 times and the 4-syllable words were duplicated 4 times. These words were mixed together with the 5-syllable words and the file was randomly scrambled. Such a file containing 3072 input words was created for each of the languages tested.

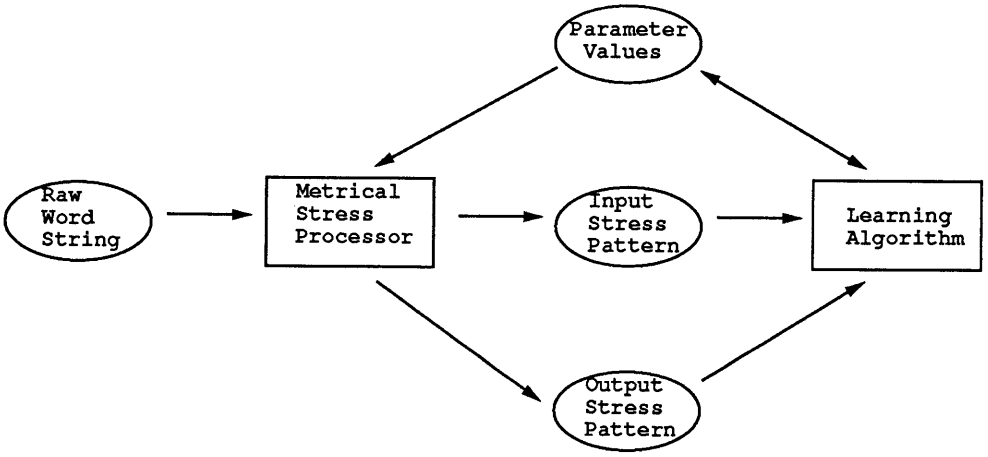

Figure 11: **Parameter-Setting Architecture for Metrical Learner**

6. Results

The learner has been tested on data from five languages: French, Maranungku, Warao, Latvian, and Lakota. This set of languages contains a mix of different stress patterns. To test the learner, 100 trials were conducted with each language. A file of 3072 randomly-ordered 3, 4, and 5-syllable words from each language was presented to the learner 100 times, and the number of input words actually processed before the learner converged to the correct hypothesis was counted. The results of testing are shown in Figure 12.

Language	Hypothesis	Average No. Examples
French	01011000000	105
Maranungku	00000000000	31
Warao	10100000010	211
Latvian	01001000000	128
Lakota	01001001000	284

Figure 12: **Test Results.** Language, parameter values, and average number of examples required for the learner to pick the correct language.

The easiest language to learn was Maranungku; because all the parameters are set to their default values in Maranungku, the initial hypothesis held by the learner (00000000000) describes the Maranungku stress pattern. The learner therefore does not need to explore any other hypotheses, and converges quickly to the initial hypothesis after 31 input examples.

French is somewhat more difficult for the learner to process, since three parameters must be set in to reach the correct hypothesis. The learner required 105 input examples to learn the fixed final stress pattern of French.

Latvian is similar to French, having fixed initial stress. Therefore it is not surprising that the learner required on average about the same number of examples (128) to learn Latvian.

Like Maranungku, Warao has a binary stress pattern; however, its otherwise smooth stress pattern is disrupted by the stress-clash avoidance rule, which changes the binary stress pattern in some cases. Since presumably stress clash does not occur in every word, it takes more examples before the learner gathers enough evidence that the stress-clash avoidance parameter (P_9) has the value 1. As a result, the learner required on average 211 examples to learn Warao. The prediction made by the model is that the presence of a stress-clash avoidance rule makes the stress pattern of a language more difficult to learn.

Of the 5 languages tested, the language that required the most examples to learn was Lakota. Lakota is like Latvian, having unbounded, left-headed feet, but with one crucial difference — in Lakota, the Extrametricality parameter (P_7) has the value 1. The left-most syllable in Lakota words is extrametrical, resulting in fixed second-syllable stress rather than fixed initial stress. The learner required 284 examples on average to learn this stress pattern. The model therefore predicts that extrametricality is hard to learn.

As implemented in this model, the 11 parameters shown in Figure 9 describe 432 possible stress systems. The learning algorithm is currently being tested on examples from more of these systems, in order to gather more empirical data about the learner's performance on a wide range of natural language data.

7. Discussion

One inherent problem in evaluating the empirical validity of phonological learners is that so little data is available on the developmental stages in phonological acquisition. It is therefore difficult to judge the relative merits of this model versus the Dresher and Kaye model, beyond stating that a model that does not enforce a rigid, predetermined order of acquisition has better hope of predicting different courses of acquisition, should they arise in further exploration of child acquisition of phonology.

The model presented here represents a number of improvements on the YOUPIE model proposed by Dresher and Kaye. The learner is incremental, i.e., it processes the input words one at a time and does not make cross-word comparisions across the entire set of data, as does YOUPIE. As a result, the learner can process large amounts of data without requiring larger and larger amounts of memory. It is therefore more plausible from a cognitive point of view, given the known limitations on child memory capacity and access to indirect negative evidence (Pinker, 1979). In addition, the complete lack of learning cues in this model implies that a) extension of the model to new formulations of UG does not require a complete reformulation and reordering of learning cues, and b) the learner does not have to perform complicated pattern matching operations on the metrical structures produced by the stress module during learning. It is also the case that YOUPIE was not exposed to ungrammatical examples during learning; in fact, the presence of ungrammatical examples example could confound the operation of the learning cues (Dresher, personal communication).

The purpose of this paper has been to show how relaxing some of the assumptions of previous parameter-setting models leads to improved behavior in certain learning situations. In particular, I presented a learning model that makes use of a uniform hypothesis weighting strategy and limited non-determinism to overcome the problems that strictly deterministic, error-driven models encounter in the presence of ungrammatical examples or when hypothesis ordering is crucial. The learning algorithm has been implemented in a system that replicates the parameters from (Dresher and Kaye, 1990) and learns the parameters of metrical phonology when presented with sets of example words. The learner has been tested on 5 languages and makes certain plausible predictions about the languages it learns. The learner also has certain desirable characteristics that make it more plausible as a cognitive model than previously proposed models of parameter setting.

8. Acknowledgements

I would like to thank Jaime Carbonell, Robin Clark, Elan Dresher, Dan Everett, Alex Franz, Ted Gibson, Prahlad Gupta, Rick Kazman, Kevin Kelly, Teruko Mitamura, Harry van der Hulst and Ken Wexler for their suggestions and comments.

9. References

Berwick Berwick, R. (1985) *The Acquisition of Syntactic Knowledge*, Cambridge: MIT Press.

Borer, H., and Wexler, K. (1987) "The Maturation of Syntax," in Williams and Roeper, eds., *Parameter Setting*, Dordrecht: Reidel.

Brown, R. (1973) *A First Language: The Early Stages*, Cambridge: Harvard University Press.

Chomsky, N. (1981) "Principles and Parameters in Syntactic Theory," in N. Hornstein and D. Lightfoot, eds., *Explanation in Linguistics: The Logical Problem of Language Acquisition*, New York: Longman.

Chomsky, N. (1985) *Knowledge of Language*, new York: Praeger.

Clark, R. (1988) "The Problem of Causality in Models of Language Learnability," Proceedings of the 13th Annual Boston University Conference on Language Development, Boston, MA.

Clark, R. (1990). "Some Elements of a Proof for Language Learnability," ms., Université de Genève.

Dresher, B. E. and J. D. Kaye (1990). "A Computational Learning Model for Metrical Phonology," *Cognition*, 34:137-195.

Flynn, S. (1987) *A Parameter-Setting Model of L2 Acquisition: Experimental Studies in Anaphora*, Dordrecht: Reidel.

Halle, M., and R. Vernaud (1987) *An Essay on Stress*, Cambridge: MIT Press.

Hyams, N. (1986) *Language Acquisition and the Theory of Parameters*, Dordrecht: Reidel.

Jaeggli, O. and K. Safir, eds. (1989) *The Null Subject Parameter*, Dordrecht: Kluwer.

Koopman, H. (1983) *The Syntax of Verbs: From Verb Movement Rules in the Kru Languages to Universal Grammar*, Dordrecht: Foris.

Nyberg, E. (1987). "Parsing and the Acquisition of Word Order," *Proceedings of the 1987 Eastern States Conference on Linguistics*, Columbus, OH, October.

Nyberg, E. (1989) "Weight Propagation and Parameter Setting: A Computational Model of Language Learning," PhD Proposal, Carnegie Mellon University, December.

Pinker, S. (1979) "Formal Models of Language Acquisition," *Cognition*, 7:217-283.

Travis, L. (1984) *Parameters and Effects of Word Order Variation*, unpublished MIT PhD thesis.

Wexler, K., and Culicover, P. (1980) *Formal Principles of Language Acquisition*, Cambridge: MIT Press.