Bootstrapping Syntax from Morpho-Phonology

Thomas R. Shultz, Vincent G. Berthiaume, and Fréderic Dandurand, *Member, IEEE*

Abstract— **It has been a puzzle how the syntax of natural language could be learned from positive evidence alone. Here we present a hybrid neural-network model in which artificial syntactic categories are acquired through unsupervised competitive learning due to grouping together lexical words with consistent phonological endings. These relatively large syntactic categories then become target signals for a feed-forward error-reducing network that learns to pair these lexical items with smaller numbers of function words to form phrases. This hybrid model learns phrasal syntax from positive evidence alone, while covering the essential findings in recent experiments on adult humans learning an artificial language. The model further predicts generalization to novel lexical words (exceptions) from knowledge of function words.**

Index Terms— **Linguistic bootstrapping, competitive learning, sibling-descendant cascade-correlation.**

I. Introduction

It has long been a mystery how the syntax of natural language can be learned from positive examples alone, without any explicit negative evidence about grammaticality. It was argued that, if a language learner ever guesses a larger, more flexible syntax, she cannot be corrected to narrow her grammar without negative evidence [1]. While the young child's utterances are sometimes corrected for truth value, they are rarely corrected for syntactic validity. Particularly vulnerable to this line of argument are error-driven learners, such as many neural-network algorithms, that would seemingly require both negative and positive to learn to distinguish ungrammatical from grammatical utterances. One solution is to assume that significant amounts of syntax are innately provided [1].

In this paper, we show that a hybrid neural-network system, combining unsupervised competitive learning (CL) with error-correcting learning (using sibling-descendant cascade-correlation, SDCC) can learn an artificial syntax from positive evidence alone. In brief, syntactic categories are acquired through unsupervised CL by grouping together lexical words that have consistent phonological endings. These relatively large syntactic categories then become target signals for a feed-forward error-reducing SDCC network that learns to pair these lexical items with smaller numbers of function words to form syntactic phrases. This hybrid system learns phrasal syntax from positive evidence alone, while covering the essential regularities in recent experiments on adult humans learning an artificial language. The system effectively bootstraps knowledge of syntactic categories and phrasal relations from the morpho-phonology of lexical words. The model further predicts generalization to novel, exceptional lexical words from knowledge of the function words alone. We begin by reviewing relevant linguistic and psychological evidence and the CL and SDCC algorithms before presenting our results.

Manuscript received March 5, 2010. This work was supported in part by a grant and a scholarship from the Natural Sciences and Engineering Research Council of Canada to TRS and VGB, respectively.

Thomas R. Shultz is with the Department of Psychology and School of Computer Science, McGill University, Montreal, QC Canada (514-398-6139; e-mail: thomas.shultz@mcgill.ca).

Vincent Berthiaume is with the Department of Psychology at McGill University, Montreal, QC Canada (e-mail vincent.berthiaume@mail.mcgill.ca).

Fréderic Dandurand is with the Laboratoire de Psychologie Cognitive, CNRS, Aix-Marseille University, Marseille, France (e-mail frederic.dandurand@gmail.com).

II. Linguistic Considerations

It is difficult to imagine how syntactic relations, such as noun phrases and verb phrases, might be acquired or even used without assuming that the relevant syntactic categories, such as nouns and verbs, are already in place. Yet syntactic categories are not typically assumed to be innately provided to language learners [2].

It is possible though that the myriad statistics in natural language might provide valuable clues for identifying syntactic lexical categories, like nouns and verbs. Two such clues involve word endings or combinations with function words [3].

Instances of a syntactic category often have common endings. For example, nouns in English are often pluralized by adding an *–s* suffix to the singular form of the noun. Likewise, English verbs are often used in the past tense by adding the suffix *–ed* to the verb stem.

Cues to noun and verb phrases are often provided by the function words that typically accompany nouns and verbs, respectively. In English, a function word often precedes a lexical one. Such function words are typically few in number compared to the more numerous lexical words. For instance, a relatively small number of determiners, such as articles (e.g., *an*, *a*, *the*) and possessive pronouns (e.g., *his*, *your*, *their*, *whose*), precede nouns, thus creating noun phrases: *the money*, *her wealth*, etc. In a similar fashion, a relatively small number of auxiliaries (e.g., *will*, *shall*, *might*, *can*, *could*, *must*, *should*) precede English verbs to express shades of time and mood in verb phrases: *can win*, *might lose*, *should continue*, etc. In comprehension, on hearing a determiner, one might well expect a following noun. Hearing an auxiliary, one could instead expect a verb.

Syntactic violations can be designed by clever psycholinguists who instead combine determiners with verbs

(*the fought*, *their lose*) or auxiliaries with nouns (*will desk*, *may projector*). Such violations would sound jarring to skilled English listeners, who might even classify them as ungrammatical phrases. Such linguistic considerations form the basis for interesting psychological experiments on the learning of artificial languages.

III. Psychological Evidence

A. Artificial Language Learning

Some of the best psychological data on statistical constraints and their relation to the problem of learning with only positive evidence comes from the learning of artificial languages. Artificial languages can be used to effectively study the properties, biases, and capacities involved in human language learning, in both infants and adults [2]. Artificial languages also present significant advantages for computational modelers of language acquisition. Modelers do not have to speculate or argue about experiences of the learners and how to measure their language skills. Instead, modelers can present exactly what was presented in the psychology experiment that is being modeled, and measure performance as it was measured in that experiment.

B. The Lany et al. Experiments

Relevant experiments employed artificial languages with an *aX*, *bY* grammar [3]. The *a* and *b* categories had only two words each, while the larger *X* and *Y* categories had six words each, five of which were used in training. Adult English speakers listened to 20 phrases of an artificial language, created by combining 5 *X* words with 2 *a* words and combining 5 *Y* words with 2 *b* words. Each phrase took 1.7s to say, with pauses of 1s between phrases and 10ms between words.

As can be seen from the examples in Table 1, the *a* words *ush* and *dak* go with *X* words, those ending with the suffix *–it*. The *b* words *ong* and *rud* accompany *Y* words, those ending in *–ul*. Sound cues to a word's category are known to facilitate the learning of such grammars [4, 5].

TABLE I
Phrases Used in Some of the Psychology Experiments

	a_1X	a_2X	b_1Y	b_2Y
Train	ush keerit		ong bivul	
Train		dak lepit		rud choopul
Train	ush feegit	dak feegit	ong habbul	rud habbul
Train	ush soolit	dak soolit	ong jerul	rud jerol
Train	ush yohvit	dak yohvit	ong pogul	rud pogul
Train	ush zamit	dak zamit	ong vummul	rud vummul
GH	ush soolit	dak zamit	ong vummul	rud pogul
GUH	ush lepit	dak keerit	ong choopul	rud bivul
NG	ush vummul	dak pogul	ong soolit	rud zamit
NG	ush choopul	dak bivul	ong lepit	rud keerit

In three different conditions of the Lany et al. study, participants were trained for 18 blocks, or 6 blocks, or in a transfer experiment involving 18 blocks with one version of a grammar followed by 6 blocks of training with the same grammar but with novel words.

After training was completed, the subjects were tested on grammatical heard phrases (GH) that were used in training, grammatical unheard phrases (GUH) that preserved the trained grammatical relations but were not actually heard in training, and non-grammatical phrases (NG) that violated the trained grammar by presenting *aY*, *bX* combinations.

During testing, subjects were asked to press the Y or N key on a computer to indicate endorsement of whether each phrase was grammatical or not, respectively. They had been told that one-half of the phrases would be grammatical and the other half not grammatical, a notice which they subsequently often ignored in their judgments.

Human results from this basic experiment are re-plotted in Figures 4, 6, and 8 for the 18-block, 6-block, and transfer conditions, respectively. In each case, we plot 1 minus the mean endorsement rate, along with SE bars. We use 1-endorsement rate here because this facilitates comparison to our simulation measure of network error. Both 1-endorsement rate and network error increase with lack of familiarity. Placing these figures just ahead of our simulation results facilitates comparison of simulation to human data.

Fig. 4 indicates that 18 blocks is enough time to learn the grammar and thus distinguish G from NG phrases. Subjects are less likely to endorse NG phrases than GUH phrases, and less likely to endorse GUH phrases than GH phrases. This suggests that they learned the phrasal grammar as well as the particular words used in training.

Fig. 6 reveals that subjects trained for only 6 blocks are less likely to endorse NG phrases and GUH phrases than GH phrases. This shows that they learned the words they were trained on, but failed to abstract the grammar underlying the phrases. The lack of difference between GUH and NG confirms that they did not learn the underlying grammar.

Fig. 8 shows that, in the transfer condition, subjects are less likely to endorse NG phrases than GH phrases, implying that they have learned the grammar. The lack of difference between GH and GUH phrases shows that it does not matter whether they heard those exact phrases in training. In other words, grammar abstraction is facilitated by training on different sets of words.

The authors conclude that their results demonstrate that sufficient exposure to positive examples of these simple grammars enables generalization to novel grammatical pairings as long as the *X* and *Y* words have distinctive and regular endings. The authors intuitively predict, but do not demonstrate, that knowing that a function word (e.g., an *a* word) is paired with some novel word, one could conclude that the novel word is likely to be an *X* word, even if it doesn't have the distinctive ending of other *X* words. Later, we demonstrate that this prediction is confirmed by our networks.

Our computational work suggests that an initial key step must also occur. Namely, learning to detect the *X* and *Y* categories based on their regular and distinctive sounds at the ends of lexical words. Once that first step is accomplished, learners might form associations between *a* and *b* function words and the morpho-phonological features that cue the category memberships of the lexical words. Eventually, generalization to novel and even exceptional lexical words

should be possible, without the distinctive sounds of the *X* and *Y* words, as we later show with our network model.

IV. Our Model

A. The CL Algorithm

We use CL to learn to sort *X* and *Y* words into two separate categories based on their distinctive and regular suffixes. CL is a special case of Kohonen's self-organizing maps, but with a very small neighborhood (consisting of only the winning output unit) and a constant learning rate (rather than a rate that decreases over learning) [6]. As shown in Fig. 1, two-word phrases are presented to both the CL and SDCC networks. Knowing ahead of time that there are two categories of interest, *X* and *Y*, we create two output units for CL.

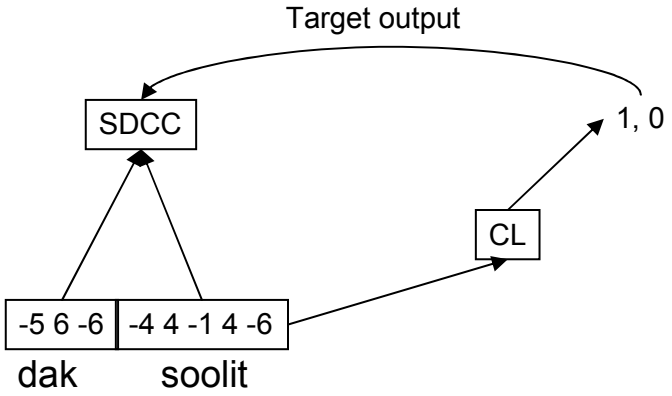

Fig. 1. Schematic of network training. Sonority-coded phrases are presented as inputs to the CL and SDCC networks. For each such input phrase, the CL network learns to output a binary signal that classifies the lexical *X* and *Y* words based on their consistent endings, and serves as target output for the SDCC network. The SDCC network uses the discrepancy between target and actual outputs to learn to classify phrases as *aX* or *bY*. Eventually, the SDCC network can use the function words (*a* or *b*) alone to classify phrases with novel and even exceptional *X* and *Y* words as *aX* or *bY* phrases.

By chance, before any weight training, given a phrasal input pattern, one output will be most active. That output unit is designated as the winner and given an activation value of 1. Weights entering the winner output are strengthened, so that this and similar input patterns will be even more likely to activate this winning output unit in the future. Another input pattern may by chance activate this output unit or a different output unit. Whichever output unit is the winner is treated in the same fashion. Non-winners are accorded activations of 0.

Kohonen's formal learning rule is quite simple:

If output $O_{up} = 1$, $\Delta w = r\left(\vec{p'} - \vec{w}\right)$

Otherwise, if output $O_{up} = 0$, $\Delta w = 0$ (1)

In words, when the activation of output unit *u*, given input pattern *p*, is 1, change the weights between the input units and output *u*; when that output activation is 0, do not change the weights. When weights are to be changed, subtract the current weight vector *w* from the input pattern vector *p*, and multiply the resulting difference vector by a smallish learning rate *r*.

This is a winner-take-all scheme, functionally equivalent to, but mathematically more elegant than, using mutually inhibitory weights between outputs. With such adjustments, the weights to winning units will eventually match the corresponding input patterns. Learning stops when the total distance between each input pattern and its closest output unit becomes less than some criterion, here .05. Various distance metrics can be used; we use Euclidean distance.

Progress of CL is shown in Figures 2 and 3. Fig. 2 shows vectors representing words ending in *–it* and those ending in *–ul*. Output units are initially placed in the center of the vowel and consonant distributions, illustrated by the stacked diamond symbols. Over 4 learning epochs, the output units migrate to the center of their respective input clusters. To increase variation in the output units and potentially simulate imperfect perception, the inputs were randomized by adding or subtracting up to .4 of their values. The rapid decline of distance over these epochs is shown in Fig. 3.

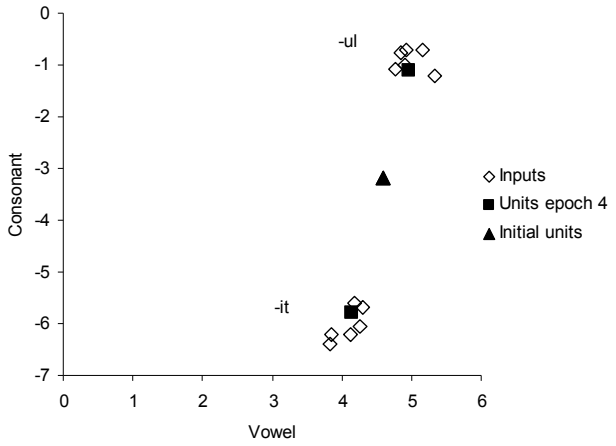

Fig. 2. Example of CL of *aX* and *bY* phrases. Two initially centered and stacked output units migrate to the regions of their respective input patterns over four training epochs. Input patterns are coded on the sonority scale.

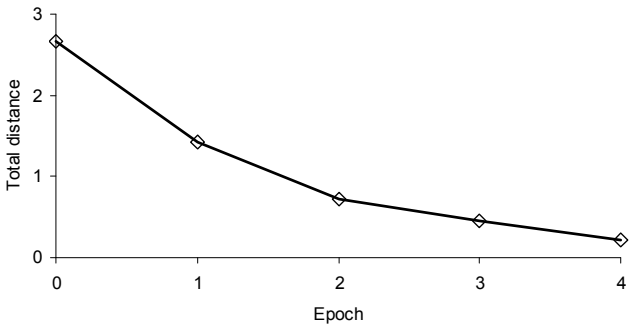

Fig. 3. In a CL network, total Euclidean distance from input patterns to their closest output unit declines over four training epochs.

B. The SDCC Algorithm

SDCC is a supervised, feed-forward, constructive neural network algorithm that adjusts connection weights to reduce network error, the discrepancy between a target vector and a vector of actual network outputs [7]. Mathematical details of SDCC and other, related cascade-correlation algorithms are well known and available in several sources [8-10].

Here we use SDCC to learn to classify phrases into the *aX*

or *bY* categories, with the output of the CL network providing the target output value used to compute network error, as illustrated in Fig. 1. For the results reported here, we translate the binary 0 or 1 CL outputs into target values of -.5 or .5, to suit the sigmoid outputs units used by SDCC.

We ran 20 networks in each of the three conditions, the same as the number of participants in the psychology experiments [3]. Each network learns a somewhat different solution because it starts with randomly initialized weights.

C. Input Coding

To code input to the CL and SDCC networks, we converted IPA values of all phonemes used in phrases to a sonority scale. As shown in Table II, sonority represents the vowel likeness of a phoneme on a numerical scale ranging from 6 (for low vowels) to -6 (for voiceless stop consonants), with more ambiguous semi-vowels and laterals pegged at -1. Such scales have a long tradition of use in phonology, psycholinguistics, and more recently in computational modeling. We used this particular scale successfully in simulations of the learning of word stress [11] and simple artificial grammars [12, 13].

TABLE II
SONORITY SCALE FOR CODING PHRASES

Phoneme category	IPA examples	Sonority
low vowels	/a/ /æ/	6
mid vowels	/Є/ /e/ /o/ /ɔ/	5
high vowels	/I/ /i/ /U/ /u/	4
semi-vowels, laterals	/w/ /y/ /l/	-1
nasals	/n/ /m/ / ŋ /	-2
voiced fricatives	/z/ /ʒ/ /v/	-3
voiceless fricatives	/s/ /ʃ/ /f/	-4
voiced stops	/b/ /d/ /g/	-5
voiceless stops	/p/ /t/ /k/	-6

In those few cases where sonority could not distinguish key phonemes, we used place of articulation (front vs. back) to make the required distinctions. An example of coding one phrase is presented in Fig. 1.

V. RESULTS

To simulate the 18-block condition, we trained SDCC networks until their output activations fell below the default score-threshold of 0.4 for all training patterns. Mean training epochs was 13.2. Network error on the test patterns after training was subjected to a repeated measures ANOVA, yielding a main effect of test pattern, $F(2, 38) = 11.06$, $p < .001$. Means and SE bars are presented in Fig. 5. Dependent-*t* tests showed more error to NG phrases than GUH phrases, $t(19) = 3.18$, $p = .005$, and marginally more error to GUH phrases than GH phrases, $t(19) = 2.05$, $p = .054$. This is the same pattern as in humans in Fig. 4, and it reflects learning of both the grammar and the trained words. Training a bit less deeply would likely take the *p*-value for that last comparison to < .05. The signature of thorough learning in both networks and people is a linear increase in detection of ungrammaticality over test patterns, as plotted here from left to right.

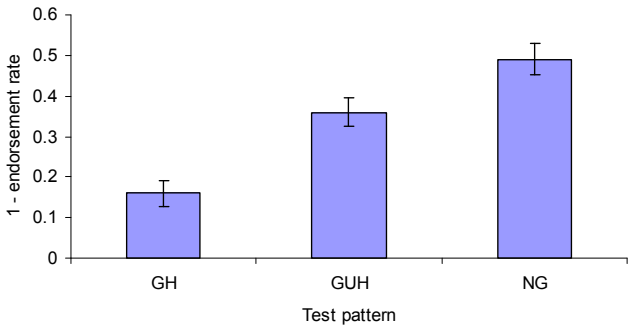

Fig. 4. Mean 1-minus-endorsement rates in adults in the 18-block condition, with SE bars (data from Lany et al., 2007). Subjects are less likely to endorse non-grammatical phrases than grammatical phrases, and less likely to endorse grammatical unheard phrases than grammatical heard phrases. This suggests that they learned the phrasal grammar as well as the words used in training.

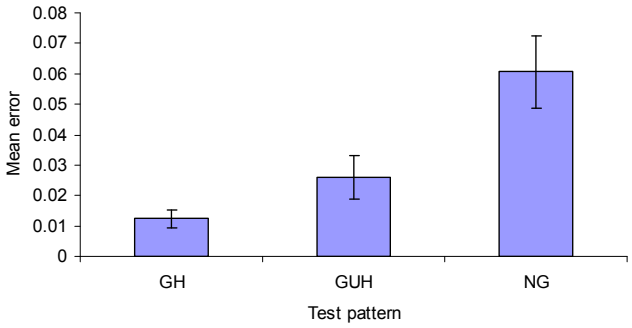

Fig. 5. Mean error in SDCC networks, with SE bars, after a mean of 13.2 training epochs. Networks show the same pattern as humans in Fig. 4: more error to non-grammatical phrases than grammatical phrases, and marginally more error to grammatical unheard phrases than grammatical heard phrases.

To simulate the 6-block condition, we trained networks for about 1/3 of the time of the 18-block condition, that is, up to 4 epochs. Means and SEs are plotted in Fig. 7. Again, there was a main effect of pattern, $F(2, 38) = 9.81$, $p < .001$. Dependent-*t* tests confirmed more error for GH than GUH patterns, $t(19) = 3.44$, $p = .003$, but no difference between GUH and NG patterns, $t(19) < 1$, *ns*. As with humans (see Fig. 6), the relatively shallow learning in this condition enables word learning but not grammar learning.

To simulate the transfer condition, we trained SDCC networks fully (until score-threshold) on one version of the grammar (for a mean of 79 epochs) and then trained them further on 4 epochs of the alternate version (with novel words). Again, the networks showed the identical pattern as humans (compare Figures 8 and 9). There was a main effect of test pattern, $F(2, 38) = 32.25$, $p < .0001$, more error to NG than GUH patterns, $t(19) = 5.00$, $p < .0001$, and no difference between GH and GUH patterns, $t(19) = 1.83$, *ns*. Training on one grammar shows transfer between the two sets of words, and the word-familiarity effect disappears. Transfer seems to be an effective way to abstract the grammar.

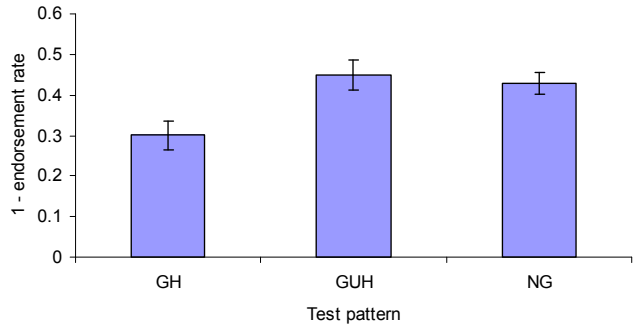

Fig. 6. Mean 1-minus-endorsement rates in adults in the 6-block condition, with SE bars (data from Lany et al., 2007). Subjects are less likely to endorse non-grammatical phrases and grammatical unheard phrases than grammatical heard phrases. This shows that subjects learned the words they were trained on, but did not abstract the grammar underlying the phrases.

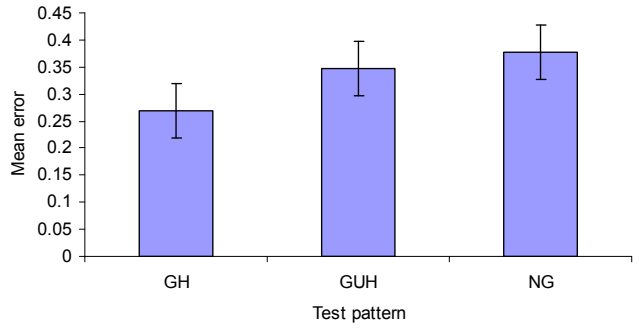

Fig. 7. Mean error in SDCC networks, with SE bars, after 4 training epochs. Networks show the same pattern as humans in Fig. 6: more error to non-grammatical and grammatical unheard phrases than to grammatical heard phrases.

We studied the possibility of transfer to lexical words with exceptional endings in several different ways, and they each worked well. Perhaps the most dramatic demonstration is provided by using novel un-patterned exceptional endings after full training on the grammar. The test patterns are regular lexical words and un-patterned exceptions (meaning unique suffixes for the lexical words) presented in either correct or incorrect syntax. For example, function words *ush* and *dak* are paired with either lexical words *wifoo* or *suffee*, and function words *ong* and *rud* are paired with lexical items *lemee* or *tamoo*. Mean and SE results for 20 networks are presented in Fig. 10.

A repeated-measures ANOVA revealed main effects for both grammaticality, $F(1, 19) = 16$, $p < .001$ and regularity, $F(1, 19) = 62$, $p < .001$, and no interaction. There was more error to NG than GUH test patterns for both regulars, $t(19) = 4.4$, $p < .001$ and exceptions, $t(19) = 3.3$, $p < .003$. This demonstrates that networks can generalize to novel lexical words (without the conventional suffixes) from knowledge of a function word alone. Because there are no comparable human results, this is a unique precise prediction.

Although SDCC ordinarily recruits hidden units to learn non-linear problems, hidden units were almost never recruited in these simulations. This underscores both the extent to which our hybrid system simplifies grammar learning and the advantages of using a constructive algorithm such as SDCC. If SDCC does not recruit any hidden units, it retains the perceptron structure it starts with, containing only input and output units.

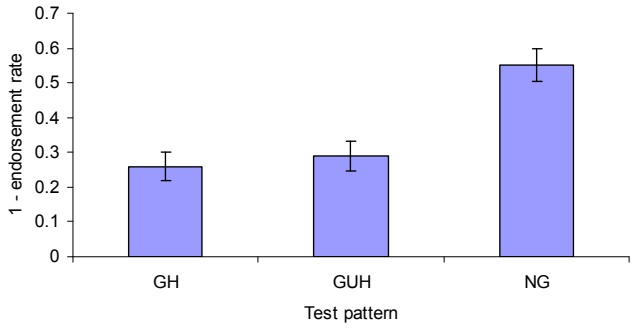

Fig. 8. Mean 1-minus-endorsement rates in adults in the transfer condition, with SE bars (data from Lany et al., 2007). Subjects are less likely to endorse non-grammatical phrases than grammatical phrases, showing that they learned the grammar.

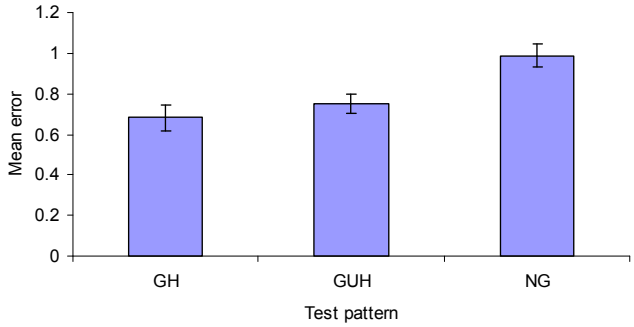

Fig. 9. Mean error in SDCC networks, with SE bars, in the transfer condition. Networks show the same pattern as humans in Fig. 8: more error to non-grammatical phrases than to grammatical phrases.

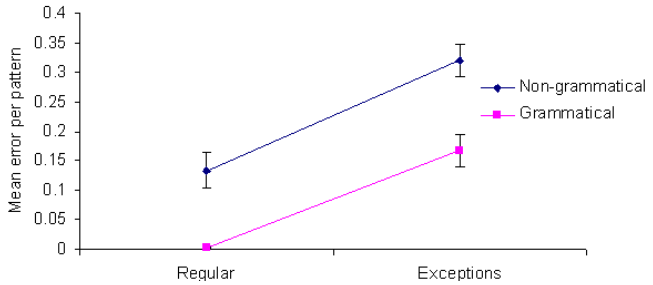

Fig. 10. Mean error in SDCC networks, with SE bars, to regular *X* and *Y* words and un-patterned exceptions presented in grammatical (*aX*, *bY*) or ungrammatical (*aY*, *bX*) syntax. Networks generalize to novel *X* and *Y* words by using the *a* and *b* words.

VI. Discussion

Our results show how syntax can be learned from positive evidence alone. For these simple phrasal grammars, syntactic categories can be acquired through unsupervised competitive learning due to grouping together lexical words that have regular phonological endings. These relatively large syntactic categories can then become target signals for a feed-forward error-reducing network that learns to pair these lexical items with smaller numbers of function words to create syntactic phrases. No negative evidence indicating the incorrectness of syntactic violations is required. Our hybrid connectionist

model learns phrasal syntax from positive evidence alone, while covering the essential regularities in experiments on adult humans learning an artificial language [3].

Our model shows similar signatures to humans across language-learning conditions. The signature of thorough learning with one set of words is a linear increase in detection of non-grammaticality across GH, GUH, and NG test patterns, suggesting learning of both the grammar and the words. The signature of superficial learning is more detection of non-grammaticality in GUH and NG test patterns than in GH test patterns with no difference between GUH and NG patterns. Superficial learning yields knowledge of words but not the grammar. Finally, transfer of a grammar from one set of words to another has a signature of more detection of grammar deviations in NG patterns than in G patterns, whether heard or not, with deep abstraction of the grammar.

Our model also predicts generalization to novel lexical words (that is, exceptions) from knowledge of the function words alone. Knowing, for example, that a novel lexical item follows a well-known article (e.g., *the*) allows a skilled listener to infer that the novel word is a noun in a noun phrase. This prediction should be tested in artificial language learning experiments with human infants and adults.

Our model thus suggests that at least some aspects of grammar can be bootstrapped from the morpho-phonology of words. Noticing consistent sounds in particular places in words can facilitate induction of syntactic categories, which in turn can facilitate learning of syntactic relations. It would be difficult to learn and use syntactic relations without being able to identify the syntactic categories that are being related.

This morpho-phonological bootstrapping of syntactic categories could complement proposals for semantic bootstrapping of syntax. It was proposed that semantic categories such as *agent* and *action* could be used to identify syntactic categories such as *subject* and *predicate*, respectively [14, 15].

Our modeling exposes possible learning mechanisms for morpho-phonological bootstrapping and provides a candidate solution to the problems of lack of negative evidence in language acquisition. Ordinarily, for supervised learning to work, both positive and negative samples are required – not only grammatically well-formed expressions but also grammatical violations that are identified as such.

Future work will likely concern simulating the remaining experimental combinations used with humans [3], converting network error into a measure closer to endorsement rate in humans, analyzing the knowledge acquired by networks, allowing CL and SDCC learning to occur in parallel, and extensions into other, more complex aspects of syntax.

Our bootstrapping approach could be compared to other proposed computational solutions to learning from positive evidence alone, such as assuming the smallest grammar [16] or most probable grammar [17] consistent with the evidence. Given the complexity of language and the rapidity with which it is acquired in humans, it is quite possible that a variety of constraints are employed. Developmental roboticists may well be interested computational models of how humans succeed in this task.

ACKNOWLEDGMENT

We are grateful to Jill Lany, Rebecca Gomez, and LouAnn Gerken for stimulating discussion and providing us with their human data to reanalyze.

REFERENCES

[1] S. Pinker, "Language acquisition," in *Language: An invitation to cognitive science*, 2nd ed. vol. 1, L. R. Gleitman and M. Liberman, Eds. Cambridge, MA: MIT Press, 1995, pp. 135-182.

[2] R. L. Gómez and L. A. Gerken, "Infant artificial language learning and language acquisition," *Trends in Cognitive Sciences,* vol. 4, pp. 178-186, 2000.

[3] J. Lany, R. L. Gomez, and G. L., "The role of prior experience in language acquisition," *Cognitive Science,* vol. 31, pp. 481-507, 2007.

[4] L. Frigo and J. MacDonald, "Properties of phonological markers that affect the acquisition of gender-like subclasses," *Journal of Memory and Language,* vol. 39, pp. 448-457, 1998.

[5] R. L. Gomez and L. LaKusta, "A first step in form-based category abstraction in 12-month-old infants," *Developmental Science,* vol. 7, pp. 567-580, 2004.

[6] T. Kohonen, *Self-organizing maps*. New York: Springer-Verlag, 1997.

[7] S. Baluja and S. E. Fahlman, "Reducing network depth in the cascade-correlation learning architecture.," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA Technical Report CMU-CS-94-209, 1994.

[8] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in neural information processing systems 2* D. S. Touretzky, Ed. Los Altos, CA: Morgan Kaufmann, 1990, pp. 524-532.

[9] T. R. Shultz, *Computational developmental psychology*. Cambridge, MA: MIT Press, 2003.

[10] T. R. Shultz and S. E. Fahlman, "Cascade-Correlation," in *Encyclopedia of Machine Learning*, C. Sammut, Ed. Heidelberg, Germany: Springer-Verlag, in press.

[11] T. R. Shultz and L. A. Gerken, "A model of infant learning of word stress.," in *Proceedings of the Twenty-seventh Annual Conference of the Cognitive Science Society* Mahwah, NJ: Erlbaum, 2005, pp. 2015-2020.

[12] T. R. Shultz and A. C. Bale, "Neural network simulation of infant familiarization to artificial sentences: Rule-like behavior without explicit rules and variables.," *Infancy,* vol. 2, pp. 501-536, 2001.

[13] T. R. Shultz and A. C. Bale, "Neural networks discover a near-identity relation to distinguish simple syntactic forms," *Minds and Machines,* vol. 16, pp. 107-139, 2006.

[14] M. Bowerman, "Inducing the latent structure of language," in *The development of language and language researchers*, F. S. Kessel, Ed. Hillsdale, NJ: Erlbaum, 1988.

[15] S. Pinker, *Language learnability and language development*: Harvard University Press, 1984.

[16] R. C. Berwick, *The acquisition of syntactic knowledge*. Cambridge, MA: MIT Press, 1985.

[17] N. Chater, "What can be learned from positive evidence?," *Journal of Child Language,* vol. 31, pp. 915-918, 2004.