April 8, 2019

Learning Reduplication with a Variable-Free Neural Network

Brandon Prickett
Aaron Traylor, *Brown University*
Joe Pater

Manuscript – 4/8/2019

Learning Reduplication with a Variable-Free Neural Network

Brandon Prickett, Aaron Traylor, and Joe Pater

bprickett@umass.edu, aaron_traylor@brown.edu, pater@linguist.umass.edu

Abstract

Reduplicative linguistic patterns have been used as evidence for explicit algebraic variables in models of cognition. Here we show that a variable-free neural network can model these patterns in a way that predicts observed human behavior. Specifically, we successfully simulate the three experiments presented by Marcus et al. (1999), as well as Endress et al.'s (2007) partial replication of one of those experiments. We then explore the model's ability to generalize reduplicative mappings to different kinds of novel inputs. Using Berent's (2013) *scopes of generalization* as a metric, we find that the model matches the scope of generalization that has been observed in humans. We argue that these results challenge past claims about the necessity for symbolic variables in models of cognition.

1. Introduction

Identity-based patterns in language have been used as evidence for explicit, algebraic variables in models of cognition (Berent, 2013; Marcus, 2001). Marcus et al. (1999) demonstrated humans' ability to learn an identity relationship by training infants on reduplicative linguistic patterns of the form ABB and ABA, where A and B were nonce words made up of a single syllable each. Marcus et al.'s (1999) participants heard a series of "sentences" made up of such words (e.g. "li na na" or "ga ti ti") and were then tested on two kinds of novel stimuli: sentences that conformed to the repetition-based pattern in the training phase and sentences that did not. The infants listened longer to novel stimuli that did not conform to the pattern they were trained on than novel stimuli that did. This was taken as evidence that the subjects had correctly learned the reduplicative pattern.

Marcus et al. (1999) demonstrated that a simple recurrent neural network (SRN; Elman, 1990) could not learn this pattern in a generalizable way, given the data that the infants were exposed to in the experiment. They attributed this failure to a lack of explicit, algebraic variables in the model. An example of a variable-based analysis of the ABB pattern would be a mapping like $\alpha\beta_1 \rightarrow \beta_2$, where α and β demonstrate syllable identity and the subscripts represent two occurrences of identical syllables. A representation like this would be blind to individual differences within the syllables and would generalize to any kind of novel stimulus. Since the infants in the experiment did generalize the pattern to novel items, and the variable-free SRN did not, Marcus et al. (1999) concluded that algebraic variables were necessary to explain their results.

A number of attempts were made to simulate the results of the experiment without using such variables (see Endress et al., 2007; Shultz & Bale, 2001 for a summary). The majority of these attempts have been dismissed because they either failed to produce a model that discriminated between novel conforming and nonconforming items or because the model used a mechanism that was equivalent to algebraic variables. These failures to simulate the results with variable-free models have been taken as further evidence that a symbolic account of cognition is necessary (Marcus, 2001).

However, a number of novel architectures and training techniques for neural networks have been developed since Marcus et al. (1999) first reported their results. The Sequence-to-Sequence network (Seq2Seq Sutskever et al., 2014) is one such architecture that has been shown to yield predictions that

correlate well with human behavior on a number of other linguistic tasks (Kirov, 2017; Kirov & Cotterell, 2018). Our work shows that a variable-free Seq2Seq network, when trained in the correct way, can successfully model Marcus et al.'s (1999) results. We then explore the model's ability to generalize to different kinds of novel items, using Berent's (2013) *scopes of generalization* as a metric for the model's success. We argue that its generalization matches that which has been observed in humans.

2. Background

The debate between connectionist and symbolic theories of language has often focused on the domain of morphology (for example, see Rumelhart & McClelland, 1986; Pinker & Prince, 1988). This includes reduplication, where all or part of a word is copied to convey some change in semantic information. Corina (1991) and Gasser (1993) first modeled the reduplicative process with recurrent neural networks. Gasser found an SRN to be insufficient for the task, citing the architecture's need for "a variable of a sort" (1993, p. 6).[1] To model the process with a neural network, he instead used a feed-forward model that could discriminate between identical and non-identical pairs of syllables.

Marcus et al. (1999) sought to test how humans learned a reduplicative pattern to see whether variables were necessary to model their behavior (see Rabagliati et al., 2019 for evidence of the reliability of these results). To do this, they trained infants on a pattern that resembled natural language reduplication, in that two out of three syllables in each stimulus were copies of one another. This resulted in two experimental conditions: infants trained on AAB patterns (e.g. with sequences like "li li na") and those trained on ABB patterns (e.g. with sequences like "li na na"). After being trained on one of the two patterns, infants were tested on a variety of items that used novel syllables, as well as novel segments within the syllables. These were either pattern conforming (e.g. "wo fe fe" for the ABB condition) or pattern nonconforming (e.g. "wo wo fe" for the ABB condition).

Their results showed that infants looked in the direction of pattern nonconforming items for significantly longer than pattern conforming ones. They took this to mean that the nonconforming items were more surprising for their subjects and that the infants had correctly learned the reduplicative pattern. The final portion of their paper described simulations that they ran with an SRN in an attempt to model the generalization seen in their experiment. While they do not describe these simulations in detail, they do report that the variable-free model failed to mimic the infants' behavior and, like Gasser (1993), Marcus et al. (1999) concluded that a recurrent neural network would need variables to learn reduplication in a human-like way.

A number of attempts have been made to model these results without the use of explicit variables. Shultz and Bale (2001) summarized a collection of these attempts and laid out a number of criteria that a model must meet to properly demonstrate that variables are not necessary for modeling Marcus et al.'s (1999) results (see also Marcus, 1999). The first criterion that they described was that the model cannot be trained on any extra data that was made using an algebraic identity function. Seidenberg and Elman (1999) did not meet this criterion in their simulation of Marcus et al.'s (1999) experiment because they exposed their SRN to pretraining that mapped sequences of syllables to an indicator of whether or not each syllable was identical to its predecessor. After the model was familiarized with this identity-based information, it was able to correctly generalize a reduplicative pattern. Since there is no evidence of infants naturally receiving such training, however, this simulation failed to provide evidence for variable-free models' ability to simulate Marcus et al.'s (1999) experiment.

[1] For discussion on how to integrate variables into connectionist models, see Marcus (2001) and Smolensky and Legendre (2006).

Another example of this criterion's relevance is Alhama and Zuidema's (2018) *Incremental Novelty Exposure*. This training technique involves presenting data to a model in a way that slowly introduces it to increasing amounts of novelty over time. This forces the neural network to find a more general solution than it might otherwise be biased toward learning, and was shown to enable a neural network to model the Marcus et al. (1999) results. Unfortunately, this use of Incremental Novelty Exposure does not meet Shultz and Bale's (2001) first criterion, since whatever mechanism creates the increasingly novel data would need an explicitly algebraic set of instructions to perform its task.

Shultz and Bale's (2001) next criterion for a variable-free model was that it could not have an architecture that explicitly compares the similarity of separate points in time. Endress et al. (2007) point out that even Shultz and Bale's (2001) proposed model does not meet this criterion, since it assumes that there are dedicated, real-valued units representing each timestep in the input. Since these can act like variables over each input feature, and since they can be explicitly compared to one another in the model's hidden layer, they are no different from variables in regards to simulating the Marcus et al. (1999) results.

The final criterion that Shultz and Bale (2001) discuss is that to generalize in a human-like way, a model must have more error for pattern non-conforming test items than for the pattern conforming ones. Christiansen & Curtin (1999) failed to meet this criterion, since their model could only differentiate between these two stimulus groups in a way that assigned more error to pattern-conforming items.

Numerous other attempts have been made to model Marcus et al.'s (1999) results, however Shultz and Bale (2001) and Endress et al. (2007) show that none of them truly meet these three criteria. Endress et al. (2007) go on to discuss a successful attempt by Altmann (2002) to model the experimental results without variables, but show that Altmann's (2002) model is unsuccessful given the majority of sampled initial weightings, and that the model makes an incorrect prediction regarding different types of nonconforming test items.

3. Our model

In this section, we present the main differences between our model and the simpler recurrent network used by Marcus et al. (1999). For the documentation on the Python packages used to implement the model, see Chollet et al. (2015) and Rahman (2016). We chose to focus on Seq2Seq models because of their recent success in a number of linguistic tasks (Cotterell et al., 2016; Kirov, 2017). For example, Kirov and Cotterell (2018) showed that a Seq2Seq network could learn both regular and irregular past tense verbs with almost perfect accuracy. Additionally, when tested on novel verbs, the model's past-tense production probabilities correlated more with experimental data from Albright and Hayes (2003) than any previously proposed model. Crucially for this paper, the Seq2Seq network has no algebraic symbols built into its architecture and does not explicitly compare the similarity of any two points in time, meaning that it meats the second criterion from Shultz and Bale (2001) discussed in §2.

3.1. Seq2Seq architecture

Seq2Seq neural networks were originally designed for machine translation and have the ability to map from one string to another, without requiring a one-on-one mapping between the strings' elements (Sutskever et al., 2014). For example, a sentence like "No, I am your father" could be mapped onto the Spanish sentence "No, soy tu padre", even though the Spanish sentence has one fewer word. The model performs this mapping by having an encoder and decoder pair built into its architecture. Each member in the pair is its

own recurrent network that steps through each piece[2] of the input sequence one at a time, with the encoder processing the input string and the decoder transforming that processed data into an output string.

An illustration of this that resembles the mappings we used in our simulations in §4.1 is shown in Figure 1. Here, the encoder passes through the entire input string (i.e. the first two syllables) before transferring information to the decoder. The decoder then unpacks this information, and produces an output string (i.e. the predicted third syllable, [fe]). The Seq2Seq architecture allows these two strings to differ in their length, with the input being four segments and the output being two. In all of the simulations discussed in this paper, the encoder is bidirectional, meaning that it passes through the input string starting from both the left and right edges.

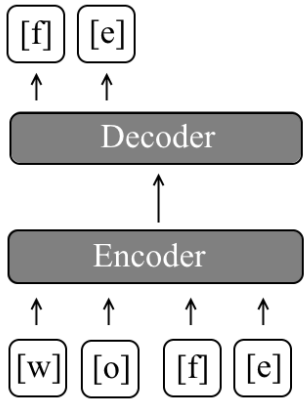

Figure 1. Illustration of Seq2Seq architecture modeling one of the stimuli in Marcus et al.'s (1999) experiments. Each box containing a transcribed sound represents a single timestep.

3.2. Long Short-Term Memory (LSTM)

LSTM (Hochreiter & Schmidhuber, 1997) is a kind of recurrent neural network layer which enhances the model's ability to store information over many timesteps. While this architectural innovation was originally designed to address the problem of vanishing gradients (Bengio et al., 1994), it has been demonstrated that LSTM can also provide models with added representational power (Levy et al., 2018).

The model performs both of these tasks by using cell states: bundles of interacting layers that can learn what information is important for the model to keep track of in the long term, and which it can forget. During training, the network is not only learning which information will allow it to predict the output from the input, but also which information at a given timestep (i.e. at a given segment in the simulations presented here) will help it to predict the output at future timesteps. While LSTM likely has some effect on our model's predictions, we leave the question of how crucial this mechanism is to future work.

3.3. Dropout

Dropout is a method used for neural networks that helps models generalize correctly to items outside of their training data (Srivastava et al., 2014). When using this method, a value between 0 and 1 is chosen that represents the probability that any given unit in the network is "dropped out" (i.e. all of its incoming/outgoing weights are temporarily set to 0), for a particular weight update during learning. This is illustrated for a single forward pass in a simple feed-forward network on the right side of Figure 2.

[2] These "pieces" of the input sequence are typically called *timesteps*. In the translation example above, each word is its own timestep, however in our simulations each timestep represents a single segment.

In this illustration, dropout causes the output units to have an activation of 2, instead of 4, because a unit in the middle layer is being dropped out and cannot contribute to the activations in the layer above it. Units are randomly sampled at each weight update during training to determine which will be dropped out and which will contribute to the model's output activations. This causes the network's solution to be more general than it might have been otherwise, since it cannot depend on a overly specific solution that uses a small number of units. For the simulations presented here, dropout was applied with equal probability to all layers of the network.

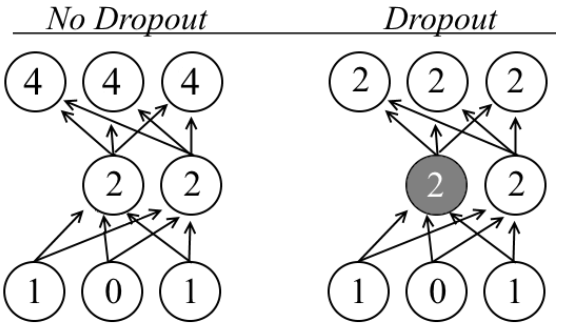

Figure 2. A simple feed-forward network, with and without dropout. Each circle is a unit and each arrow is a connection. Dropped out units are in grey. Each unit's output (before dropout) is denoted by the number inside of it. All connections have a default weight of 1 and all activation functions are f(x)=x.

4. Modeling Marcus et al. (1999)

4.1. Experiments 1 and 2

In their first two experiments, Marcus et al. (1999) demonstrated that infants could generalize ABB and ABA patterns to novel segments. To simulate this, we trained our model to predict the third syllable in each experimental item, based on the first two. For all of the simulations presented in this section, the Seq2Seq model was given a four-segment input representing the first two syllables, and asked to produce a two-segment output representing the third syllable (as illustrated in Figure 1). Segments were represented using vectors made up of 11 feature values, based on standard features used in phonological theory. These features, along with the segments from Marcus et al.'s experiments that they describe, are given in the appendix. There were also 11 units in each of the 4 hidden layers. The model was trained using RMSProp (Tieleman & Hinton, 2012), a gradual, error-based algorithm, with the default hyperparameter values used in Keras (Chollet, 2015). The loss that this algorithm minimized was mean squared error (MSE) and the probability of dropout was .85 (all hyperparameters were chosen after a small amount of pilot testing before running our final simulations).

In addition to being trained on the same items as Marcus et al.'s (1999) subjects (i.e. trisyllabic words that followed either an ABB or ABA pattern), the model also went through a pretraining phase meant to familiarize it with the syllables used in the experiment. Preliminary simulations that were run without this pretraining failed to reproduce the kind of generalization observed in the experiment. The pretraining can be thought of as simulating the experience that the infants would have had with English syllables prior to participating in the experiment (since all of the syllables that were used are attested in English). Unlike Seidenberg and Elman (1999), there was no identity-based information in this pretraining, meaning that it did not violate the first criterion laid out by Shultz and Bale (2001). Each learning datum in pretraining was a set of two randomly sampled syllables that mapped to another randomly chosen syllable. After being trained on 1,000 of these randomly produced data for 1000 epochs with batches of size 50, the model's

decoder weights were set back to their original value (with the encoder weights being preserved) and the experiment simulation began.

The model was then trained for 500 epochs (again, with batches of size 50) on a dataset that contained three copies each of the items from Marcus et al.'s (1999) training phase. A new random ordering of these data was sampled for each simulation. At the end of this training, the model was tested on a dataset that contained three copies each of the appropriate test items using Keras's "test_on_batch()" function (Chollet, 2015). We used the MSE values obtained from these tests as a dependent variable to compare to the infant listening times reported by Marcus et al. (1999). The results for 200 simulations[3] (50 per condition, per experiment) are shown below, along with the results reported by Marcus et al.'s (1999) 32 subjects (8 per condition, per experiment).

	Average MSE (SE)				**Average listening time (SE)**			
	Conf.	Nonconf.	t(99)	p	Conf.	Nonconf.	F(14)	p
Exp. 1	0.49 (0.01)	0.52 (0.01)	-2.8	<.01*	6.3 (0.65)	9.0 (0.54)	25.7	<.01*
Exp. 2	0.67 (0.01)	0.68 (0.01)	-3.3	<.01*	5.6 (0.47)	7.35 (0.68)	25.6	<.01*

Table 1. Results from our simulations and the corresponding experiments in Marcus et al. (1999). All MSE values are rounded to the nearest hundredth and averaged across runs. Values in parentheses show standard error of the mean.

The results in Table 1 demonstrate that the model, like the infants, differentiates between conforming and nonconforming items in the test data. After running paired t-tests on the MSE values, both Experiment 1 ($t[99]=-2.8$, $p=.003$) and Experiment 2 ($t[99]=-3.3$, $p=.0006$) showed significantly less MSE for conforming test stimuli than for nonconforming ones.[4] This means that the nonconforming stimuli were predicted more poorly by the model, meeting the final criterion laid out by Shultz and Bale (2001).[5]

4.2. Experiment 3

Marcus et al.'s (1999) third experiment required a different set-up than our previous simulations. In this experiment, infants were trained on either an ABB pattern or an AAB pattern. This was designed to ensure that the infants had not simply learned to expect changes across syllable boundaries in the ABA condition, and a lack of such change in the ABB condition. However, as pointed out by Endress et al. (2007), this means that the problem can no longer be modeled as a mapping from the first two syllables to the third, since the model would have no way of predicting the third syllable in AAB sequences.

To overcome this issue, we designed a new kind of simulation in which the model's input included three syllables, but the middle syllable in the input was represented by two empty segments (i.e. segments that had a value of 0 for every feature). The output of the model was a single syllable that was intended to represent the material the empty syllable was supposed to include. This is illustrated in Figure 3. Since the second syllable is predictable in both the AAB and ABB conditions, given the other two syllables, this

[3] To avoid p-hacking, we ran numerous pilot tests to gauge how many simulations were necessary to gain statistical significance. After the pilots, we reran all 200 simulations and ran all t-tests on these new results.

[4] Following Marcus et al. (1999), we combined results from both the ABB and ABA conditions in each experiment, however both groups showed qualitatively similar results.

[5] One major difference between Marcus et al.'s (1999) results and those produced by our model is their respective effect sizes. We do not find this difference troubling, since their subjects' learning could have been aided by the various instances of repetition present in their previous linguistic experience. Common examples of this would be words like "Mama" (Ferguson, 1964) or the reduplicative processes present in English morphology (Ghomeshi et al., 2004).

allowed us to test the model on a mapping that was relevant to the design of Experiment 3, while still maintaining the sequential nature of the input that the subjects would have had in the experiment.

For pretraining in these simulations, the model was trained to map two randomly chosen syllables with an empty syllable in between them to another randomly chosen syllable. After this pretraining, as in the previous simulations, the decoder's weights were set back to their initial values. To simulate the experiment's training phase, the models were then trained on a data set as described in the previous section.

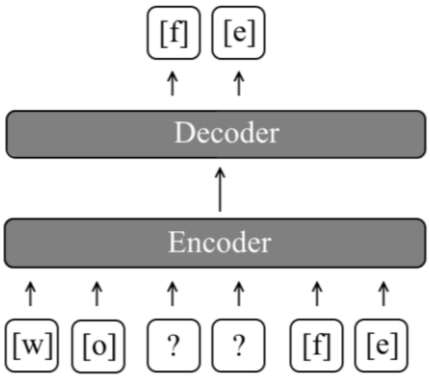

Figure 3. Illustration of mappings in Experiment 3's simulations. The "?" symbol represents an empty segment.

The test phase was identical to those described in §4.2, except for an additional kind of test item. This additional type of test was designed to simulate the AAA stimuli in Endress et al.'s (2007: Appendix A) replication of Marcus et al.'s (1999) third experiment. Endress et al. (2007) included these stimuli in the test phase to test a prediction made by Altmann's (2002) model. That model correctly predicted a preference in the model for conforming stimuli over Marcus et al.'s (1999) nonconforming ones, however it predicted an even stronger preference for stimuli that followed an AAA-style pattern. That is, stimuli such as "wo wo wo", where all three syllables are the same. Endress et al. (2007) showed that when a replication of Marcus et al.'s (1999) third experiment was run that also tested subjects' preferences for this kind of stimulus, humans still preferred items that conformed to the reduplicative pattern they were trained on. To ensure that the interpretation of our model's results does not fall into the same trap as Altmann's (2002), we tested it both on the Marcus et al. (1999) style of nonconforming stimulus (i.e. AAB for the ABB pattern and ABB for the AAB pattern) as well as the Endress et al. (2007: Appendix A) test items (i.e. AAA). The results from 20 such simulations (10 in each condition) are given in the two tables below.

These simulations show that our model can predict the results of Marcus et al.'s (1999) third experiment, as well as Endress et al.'s (2007) replication of that experiment. The model's MSE was significantly higher for both the standard nonconforming items (t[19]=-2.30, p=.01635), as well as the AAA nonconforming ones (t[19]=-2.22, p=.01933).[6]

Average MSE (SE)				**Average listening time (SE)**			
Conforming	Nonconforming	t(19)	p	Conforming	Nonconforming	F(14)	p
.56 (.01)	.57 (.01)	-2.30	.01635*	6.4 (0.38)	8.5 (0.50)	40.3	<.001*

Table 2. Results for the Experiment 3 simulation, compared to Marcus et al.'s (1999) results. Values in parentheses show standard error of the mean.

[6] The effect size for these simulations was larger than those in the previous section—however this is likely due to the larger number of test items used for this experiment (following Endress et al.'s 2007 procedure).

Average MSE (SE)			
Conforming	AAA	t(19)	p
.56 (.01)	.57 (.01)	-2.22	.01933*

Table 3. Results comparing Endress et al.'s (2007) conforming (AAB/ABB) and nonconforming stimuli (AAA). Values in parentheses show standard error of the mean.

5. Exploring the model's scope of generalization

In §4, we demonstrated our model's ability to simulate Marcus et al.'s (1999) experiment results, despite its lack of variables. However, these results only paint a partial picture of how well the model is able to generalize reduplication. Marcus et al. (1999) tested infants on words that used segments that were completely novel in the context of the experiment (i.e. they were present in the words that infants were trained on), however, all of the segments in the experiment were present in English, which means that the infants would have had a considerable amount of experience with them. We simulated this experience in our models using randomly produced pretraining, meaning that the model never needed to generalize reduplication to completely novel phonemes.

To better understand how well our model can generalize to data that it has not been exposed to at all, we structured the simulations in this section to map a single syllable (e.g. "ba") to two copies of itself (e.g. "baba").[7] We then tested how well the model generalized this mapping when given withheld data at various levels of novelty. To do this, we followed Berent's (2013) proposal regarding the *scopes of generalization* that are possible for such reduplicative patterns. We summarize the three scopes here, and then in §5.1-5.3, we explain the series of simulations we ran to determine which scope describes our model's performance.

The simplest form of generalization that Berent (2013) described is to novel syllables. This is illustrated for a reduplicative pattern in the table below, with the grey cells representing the input syllables seen in the training data and the bolded syllable being the input for a test item withheld from training.

	i	e	o	a
p	pi	pe	po	pa
b	bi	be	bo	ba
t	ti	te	to	ta
d	di	de	do	**da**

Table 4. Example of generalization to a novel syllable. Grey cells represent training data, bolded items indicate the crucial testing item.

If a model correctly predicts the mapping [da]→[dada] after being trained on data that does not include the syllable [da] (but that does include other syllables containing both [d] and [a]), it would successfully be performing this scope of generalization. This would demonstrate that the model did not simply memorize individual input+output pairs, but doesn't show that the model has learned anything more sophisticated than how to copy individual segments. For example, it could have learned patterns like "if [d] occurs as the first segment in the input, make [d] the first and third segments in the output."

The next scope is generalizing to novel segments. As mentioned in §4, we represent segments as vectors of phonological features. When testing this scope, we trained the model on every relevant value for each

[7] This also resembles natural language reduplication more closely than the Marcus et al. (1999) pattern does. For example, reduplication in the language Karao which doubles the stem of a word to change the number of some verbs (e.g. [man**ba**kal] "fight each other, 2 people" → [man**baba**kal] "fight each other, >2 people").

feature, but not on all of the possible feature value combinations. This is demonstrated in Table 5, using the same coloring scheme as above.

	i	e	o	a
p	pi	pe	po	pa
b	bi	be	bo	ba
t	ti	te	to	ta
d	di	de	do	**da**

Table 5. Example of generalization to a novel segment. Grey cells represent training data, bolded items indicate crucial testing item.

In the example above, the model is trained on syllables containing [p], [b], and [t], but never sees [d]. This would give it experience in training with all of the feature values that make up [d] (since it shares every value but [voice] with [t] and it *does* share its value for [voice] with [b]), without ever seeing them together in the same vector. This scope of generalization demonstrates that a learner is doing more than just memorizing a mapping for each segment. Instead, if a model exhibits this level of generalization, it has acquired a broader generalization that references specific feature values. For example, it might learn the generalization "if the first segment in the input is -1 for [voice], make the first and third segments in the output have a value of -1 for [voice]."

Berent (2013) points out that generalization to novel segments would still not demonstrate that a model has learned a fully algebraic function. To show this, a model would need to demonstrate its ability to generalize to novel feature values, which Berent (2013) calls "across the board" generalization. This is demonstrated in the table below, where the learner is only trained on oral consonants (i.e. sounds made without the use of the nasal cavity) and then tested on the nasal consonant [n].

	i	e	o	a
p	pi	pe	po	pa
b	bi	be	bo	ba
t	ti	te	to	ta
d	di	de	do	da
n	ni	ne	no	**na**

Table 6. Example of generalization to a novel feature value. Grey cells represent training data, bolded items indicate crucial testing item.

In the example above, the model has only been exposed to the feature value [nasal]=−1 in its input, so if it generalizes to [na], there is no way it could have learned a pattern that depends on feature-based mappings. Generalization to novel feature values means that a model has learned that the pattern is independent of any particular feature. For example, the model could have learned the function $\alpha \rightarrow \alpha\alpha$, where α can be any arbitrary string of sounds.

To test which scope of generalization our model could achieve, we ran three kinds of simulations that were more carefully aimed at this question than the Marcus et al. (1999) experiment: one in which the model was tested on a novel syllable made up of segments it had seen reduplicating in its training data (§5.1), one in which the model was tested on a syllable made with a segment that it hadn't received in training (§5.2), and one in which the model was tested on a syllable with a novel segment containing a feature value that hadn't been presented in the training data (§5.3). None of the simulations described here used a pretraining phase like those in §4.

In the results presented in this section, the set of possible segments and the feature values representing those segments were randomly produced in each simulation. Features for these simulations were binary (either -1 or 1), to avoid ambiguity in interpreting the model's success. To ensure that each language had consonants and vowels present in its segment inventory, segments were divided into consonants and vowels by treating the first feature as [syllabic], i.e. any of the randomly produced feature vectors that began with -1 were considered a consonant and any that began with 1 were considered a vowel. No randomly produced language inventories were used that consisted of only consonants or only vowels.

The toy language for any given simulation consisted of all the possible consonant+vowel syllables that could be made with that simulation's randomly created segment inventory (all inventories contained forty segments total, unless otherwise noted). Crucially, before the data was given to the model, some portion of it was withheld for testing (see the subsections below for more information on what was withheld in each testing condition). The mapping that the model was trained on took a single syllable (e.g. [ba]) as input and produce two syllables (e.g. [baba]) as output, as shown in Figure 4.

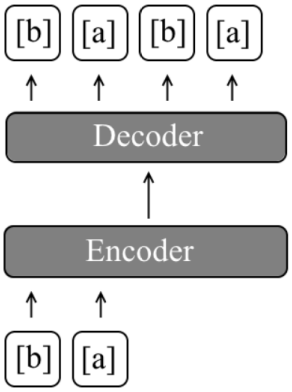

Figure 4. Illustration of mappings in this section's simulations.

The models were trained for 1000 epochs, with training batches that included all of the learning data. There were 18 units in the model's hidden layer, the probability of dropout was either 0 or .75, and all other hyperparameters were the same as in §4 (as in the previous section, hyperparameters were chosen after a small amount of piloting was performed). To test whether the model generalized to withheld data at the end of training, a much stricter definition of success was used than in the Marcus et al. (1999) experiments. The model was given the relevant withheld item as input, and the output it predicted was computed using Keras's "predict()" function (Chollet, 2015). Since the model is not probabilistic, these predictions do not vary given the same input and set of connection weights. The predictions were compared to the corresponding correct outputs (i.e. the reduplicated form of the stem it was given). If every feature value in the predicted output had the same sign (positive/negative) as its counterpart in the correct output, the model was considered to be successfully generalizing the reduplication pattern. However, if any of the feature values did not have the same sign, that model was considered to have failed at the generalization task.

5.1. Generalization to novel syllables

Our first set of simulations tested whether the model could generalize to novel syllables. If the model failed at this task, then it would mean that it was memorizing whole syllables in the training data, rather than extracting any actual pattern from the mappings it was trained on. The model successfully reduplicated all of the syllables it had been trained on in all runs for this condition. Additionally, when no dropout was used, it successfully generalized to novel syllables in 22 of the 25 simulations (88%). This shows that a standard

Seq2Seq model, with LSTM but no dropout, can perform generalization to novel syllables, and does so a majority of the time. Dropout did not have a noticeable effect on the model's ability to generalize. When the probability of units dropping out was .75, it again generalized to novel syllables in 22 of the 25 simulations (88%).

5.2. Generalization to novel segments

Our next set of simulations tested the model's ability to generalize to novel segments. If the model failed at this task it would mean that it was only learning generalizations that referred to individual sounds, such as "if [d] is the first segment in the input, make [d] the first and third segments in the output." The model successfully reduplicated syllables from training in 24 of the 25 runs for this condition when no dropout was applied. However, it failed to generalize to novel segments in the majority of runs, with only 6 out of 25 simulations being successful (24%). This shows that a standard Seq2Seq model, with LSTM but no dropout, does not reliably generalize to unseen segments.

However, when the probability of a unit dropping out was increased to .75, the model successfully reduplicated syllables from training in all runs and generalized to novel segments in 15 out of 25 runs (60%). This means that as long as dropout is used in training, the model will reliably achieve this scope of generalization. This difference between the two dropout conditions is illustrated in Figure 5.

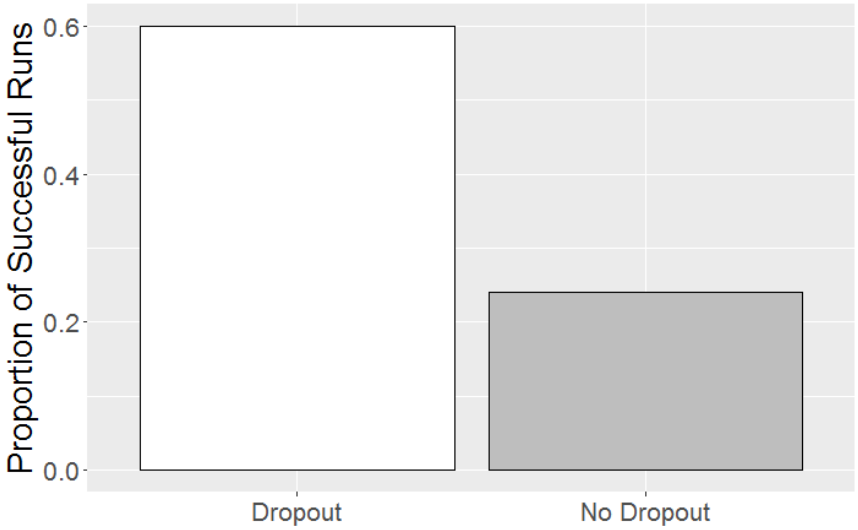

Figure 5. Difference between dropout with probabilities of .75 and 0 in generalization to novel segments.

5.3. Generalization to novel feature values

Our next set of simulations tested the model's ability to generalize to novel feature values. Failing at this means that the model learned generalizations that depend on features, rather than completely abstract algebraic functions like α→αα. For the results reported here, toy languages always contained 43 segments in their inventory. The feature vectors that represented these segments are given in Table II of the Appendix. In this condition, the inventory was designed by hand, rather than being randomly produced, and the withheld segment was always [n], with the withheld feature value being [nasal]=1. A variety of other segment inventories were tested, with no changes in the model's performance.

Despite the fact that the model achieve perfect performance on trained syllables, it was never able to generalize to novel feature values, regardless of whether dropout probability was 0 or .75. A number of other dropout settings were attempted with no success at increasing the scope of generalization to this level. This suggests that Seq2Seq models, regardless of dropout, cannot generalize to novel feature values.

6. Discussion

6.1. Summary of results

In §4, we showed that a Seq2Seq model without any explicit variables can capture the results from all three of Marcus et al.'s (1999) experiments. We also demonstrated that unlike Altmann's (2002) model, ours does not predict a preference toward AAA items when trained on AAB and ABB sequences. This means our model can also predict the results reported by Endress et al. (2007).

Next, we probed our model further in §5, more carefully testing which scope of generalization it could capture when trained on a reduplicative pattern. A summary of these results can be viewed side by side in Figure 6. The findings from this series of simulations showed that even without dropout, a Seq2Seq model is not simply memorizing mappings for each individual syllable, since it was able to generalize reduplication to novel items. We also showed that the model, when using dropout in training, can reliably generalize reduplication to novel segments.

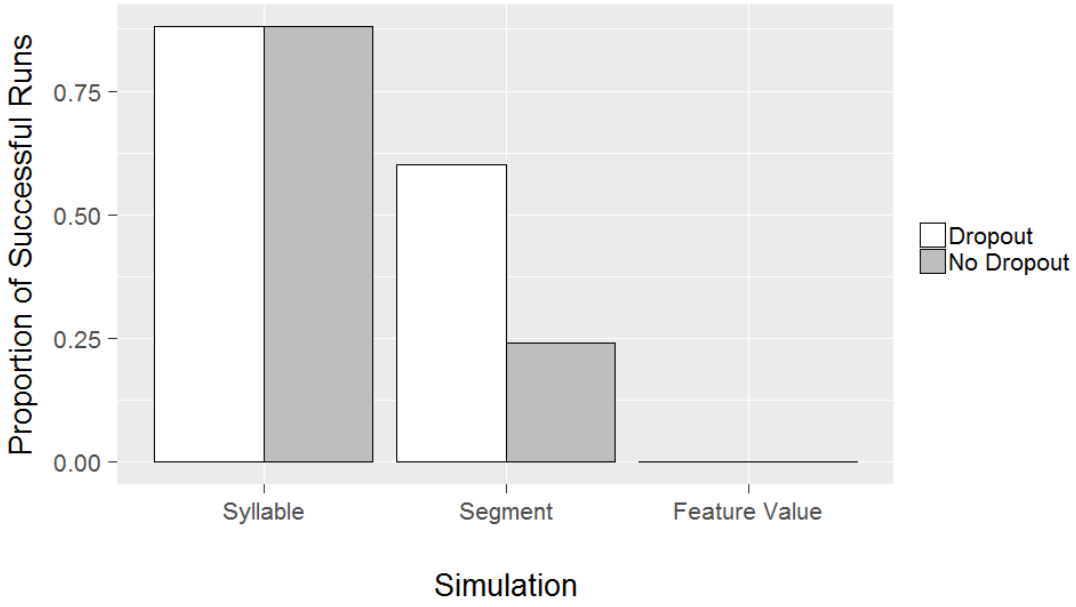

Figure 6. Summary of results for each dropout condition and scope of generalization.

6.2. Future work

There are a number of avenues that present themselves for future work. First of all, exploring more of the experimental predictions that the model makes could yield new kinds of experiment designs to explore. Additionally, running experiments on humans that test for generalization of reduplicative patterns to truly novel feature values could help determine whether that scope of generalization should be a goal for models of cognition. The learning biases inherent to this model could also yield important predictions. For example, Endress et al. (2007) found that identity-based patterns were easier to learn than patters than did not make

any reference to identity. Testing to see whether this model shows an identity bias could be another way of testing whether variables are necessary to correctly predict human learning.

While we've shown that dropout increases the Seq2Seq model's scope of generalization, it remains an open question whether other forms of regularization (such as an L2 prior) would be successful as this task. One hypothesis for why dropout is helpful is that it causes certain training data to be indistinguishable from crucial testing data. For example, if the training set includes the stems [pa] and [da], but [ta] is withheld, a model without dropout would not generalize to the novel item because it was never trained on reduplicating [t]. However, when dropout is applied, in a subset of epochs the unit activations distinguishing [t] from [d] could be dropped out. This would allow the model to learn how to reduplicate a syllable that is ambiguous between [ta] and [da]. Crucially, this would not allow the model to generalize to novel feature values that were never activated in training (such as [nasal]=1), but these ambiguous training epochs could provide enough information for generalization to withheld segments. If this hypothesis is correct, then other forms of regularization may not be as successful at increasing the model's scope of generalization. We leave testing these other methods to future research.

6.3. Can humans generalize to novel feature values?

When discussing generalization of reduplicative patterns, Berent (2013) used Hebrew speakers' judgments regarding an ABA pattern present in their language's phonotactics. In Hebrew, the first two consonants in a word's stem cannot be identical (i.e. the first three segments are not allowed to match the pattern ABA, where the B is any arbitrary vowel). For example, the word [simem] 'he intoxicated' is acceptable, while the nonce word *[sisem] is not. Berent (2013) reviewed a number of past experiments that showed speakers generalizing this pattern by having them rate the acceptability of various kinds of novel words.

The first results Berent (2013) presented were from Berent and Shimron (1997) and demonstrated Hebrew speakers generalizing to novel words (which would be equivalent to the "novel syllables" discussed above). These words were made up of segments that were attested in Hebrew, such as [s] and [m]. Speakers in this experiment rated words with s-s-m stems (like *sisem above) as significantly less acceptable than words with s-m-m and p-s-m stems. This demonstrated that Hebrew speakers were doing more than just memorizing the lexicon of their language (i.e. that they could extract phonotactic patterns).

The next results that Berent (2013) presented involved Hebrew speakers generalizing the pattern to novel segments (Berent et al., 2002). The segments of interest were /tʃ/, /dʒ/ and /w/, all of which are not present in native Hebrew words. Even when these non-native phonemes were used, Hebrew speakers rated words whose first two consonants were identical (e.g. dʒ-dʒ-r) as worse than those that did not violate the phonotactic restriction (e.g. r-dʒ-dʒ). This demonstrated that speakers had not just memorized a list of consonants that cannot cooccur (e.g. *pp, *ss, *mm, etc.) while acquiring their phonological system, since this list would not have included sounds like [w].

Finally, Berent (2013) showed that speakers can generalize the pattern to the segment [θ], which she claimed represented generalization to the novel feature [wide] (Berent et al., 2002). However, [wide] is not used in standard phonological feature theory (e.g. Chomsky & Halle, 1968; Hayes, 2011). Using a standard featural representation for [θ], such as [+anterior, +continuant, -strident], would mean that [θ] does not represent a novel feature value for Hebrew, since the language contains other, native, [+anterior], [+continuant], and [-strident] sounds (e.g. [t], [ʃ], and [f], respectively). To our knowledge, no experiment has tested humans' ability to generalize to truly novel feature values.[8] Such an experiment would be

[8] Berent et al. (2014) claim to observe generalization of a reduplication pattern in American Sign Language to novel feature values. However, they are using the word "feature" differently than we do here. While they define features as

difficult, since children stop reliably perceiving novel feature contrasts at a relatively young age (see, e.g. Werker & Tees, 1983). Because of this, we cannot conclude whether our model's results from §4.3 are human-like or not.

6.4. Conclusions

In the past, it has been claimed that it is impossible for variable-free neural networks to generalize reduplicative patterns (Berent, 2013; Marcus, 2001; Marcus et al., 1999). Here, we presented results showing that a network with no variables (that's been pretrained on randomized data) can capture Marcus et al.'s (1999) experimental results. Since our simulations met all three of the criteria laid out by Shultz and Bale (2001) for a successful variable-free model, our results challenge the claim that modeling these results is only possible with a symbolic model of cognition.

We also probed our model's generalization abilities to determine more precisely what scope of generalization it was using. We found that it could generalize to novel syllables and novel segments, but not to novel feature values. This matches the scope of generalization observed thus far in humans, and also explains why pretraining was necessary for our model to simulate Marcus et al.'s (1999) results.

More broadly, this paper challenges the idea that variable-free neural networks are insufficient for modeling human behavior and provides another example of the Seq2Seq architecture successfully mirroring the linguistic capabilities of humans.

Acknowledgments

The authors would like to thank the members of the UMass Sound Workshop, the members of the UMass NLP Reading Group, Tal Linzen, Ryan Cotterell, and the audience at the 2019 meeting of SIGMORPHON for helpful discussion and feedback. This work was supported by NSF Grant #BCS-1650957.

Declaration of Interest

We have no conflicts of interest to declare.

References

Albright, A., & Hayes, B. (2003). Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, *90*(2), 119–161.

Alhama, R. G., & Zuidema, W. (2018). Pre-Wiring and Pre-Training: What does a neural network need to learn truly general identity rules? *Journal of Artificial Intelligence Research*, *61*, 927–946.

Altmann, G. T. (2002). Learning and development in neural networks–the importance of prior experience. *Cognition*, *85*(2), B43–B50.

Berent, I. (2013). The phonological mind. *Trends in Cognitive Sciences*, *17*(7), 319–327.

Berent, I., Dupuis, A., & Brentari, D. (2014). Phonological reduplication in sign language: Rules rule. *Frontiers in Psychology*, *5*, 560.

Berent, I., Marcus, G., Shimron, J., & Gafos, A. I. (2002). The scope of linguistic generalizations: Evidence from Hebrew word formation. *Cognition*, *83*(2), 113–139.

Berent, I., & Shimron, J. (1997). The representation of Hebrew words: Evidence from the obligatory contour principle. *Cognition*, *64*(1), 39–72.

Brentari, D. (1998). *A prosodic model of sign language phonology*. Mit Press.

Chollet, F. (2015). Keras. Retrieved January 18, 2018, from https://github.com/keras-team/keras

Chomsky, N., & Halle, M. (1968). *The sound pattern of English.* New York, NY: Harper & Row.

the description of an entire hand shape, our definition is closer to the sign language features proposed by Brentari (1998), where feature values are the most atomic part of a sign's representation.

Christiansen, M. H., & Curtin, S. L. (1999). The power of statistical learning: No need for algebraic rules. In *Proceedings of the 21st annual conference of the Cognitive Science Society* (Vol. 114, p. 119). Citeseer.

Corina, D. P. (1991). *Towards an understanding of the syllable: evidence from linguistic, psychological, and connectionist* (PhD Thesis). University of California, San Diego.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., & Hulden, M. (2016). The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology* (pp. 10–22).

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Endress, A. D., Dehaene-Lambertz, G., & Mehler, J. (2007). Perceptual constraints and the learnability of simple grammars. *Cognition*, *105*(3), 577–614.

Ferguson, C. A. (1964). Baby talk in six languages. *American Anthropologist*, *66*(6_PART2), 103–114.

Gasser, M. (1993). *Learning words in time: Towards a modular connectionist account of the acquisition of receptive morphology*. Indiana University, Department of Computer Science.

Ghomeshi, J., Jackendoff, R., Rosen, N., & Russell, K. (2004). Contrastive Focus Reduplication in English (The Salad-Salad Paper). *Natural Language & Linguistic Theory*, *22*(2), 307–357.

Hayes, B. (2011). *Introductory phonology* (Vol. 32). John Wiley & Sons.

Kirov, C. (2017). Recurrent Neural Networks as a Strong Domain-General Baseline for Morpho-Phonological Learning. In *Poster presented at the 2017 Meeting of the Linguistic Society of America*.

Kirov, C., & Cotterell, R. (2018). *Recurrent Neural Networks in Linguistic Theory: Revisiting Pinker & Prince (1988) and the Past Tense Debate*.

Marcus, G. (1999). Do infants learn grammar with algebra or statistics? Response. *Science*, *284*(5413), 436–437.

Marcus, G. (2001). *The algebraic mind*. Cambridge, MA: MIT Press.

Marcus, G., Vijayan, S., Rao, S. B., & Vishton, P. M. (1999). Rule learning by seven-month-old infants. *Science*, *283*(5398), 77–80.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, *28*(1), 73–193. https://doi.org/10.1016/0010-0277(88)90032-7

Rabagliati, H., Ferguson, B., & Lew-Williams, C. (2019). The profile of abstract rule learning in infancy: Meta-analytic and experimental evidence. *Developmental Science*, *22*(1), e12704.

Rahman, F. (2016). *seq2seq: Sequence to Sequence Learning with Keras*. Python. Retrieved from https://github.com/farizrahman4u/seq2seq

Rumelhart, D., & McClelland, J. (1986). On learning the past tenses of English verbs. In J. McClelland & D. Rumelhart (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 2: Psychological and Biological Models, pp. 216–271). The MIT Press.

Seidenberg, M. S., & Elman, J. L. (1999). Do infants learn grammar with algebra or statistics? *Science*, *284*(5413), 433f–433f.

Shultz, T. R., & Bale, A. C. (2001). Neural network simulation of infant familiarization to artificial sentences: Rule-like behavior without explicit rules and variables. *Infancy*, *2*(4), 501–536.

Smolensky, P., & Legendre, G. (2006). *The harmonic mind: From neural computation to optimality-theoretic grammar (Cognitive architecture), Vol. 1*. MIT press.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*(2), 26–31.

Werker, J. F., & Tees, R. C. (1983). Developmental changes across childhood in the perception of non-native speech sounds. *Canadian Journal of Psychology/Revue Canadienne de Psychologie*, *37*(2), 278.

Appendix

Segment	[syllabic]	[sonorant]	[voice]	[coronal]	[continuant]	[labial]	[velar]	[anterior]	[high]	[low]	[back]
p	-1	-1	-1	0	-1	1	0	0	0	0	0
b	-1	-1	1	0	-1	1	0	0	0	0	0
t	-1	-1	-1	1	-1	0	0	1	0	0	0
d	-1	-1	1	1	-1	0	0	1	0	0	0
k	-1	-1	-1	0	-1	0	1	0	0	0	0
g	-1	-1	1	0	-1	0	1	0	0	0	0
f	-1	-1	-1	0	1	1	0	0	0	0	0
dʒ	-1	-1	1	1	-1	0	0	-1	0	0	0
l	-1	1	0	1	0	0	0	1	0	0	0
w	-1	1	0	0	0	1	1	0	1	-1	1
a	1	1	0	0	0	0	0	0	-1	1	1
i	1	1	0	0	0	0	0	0	1	-1	-1
o	1	1	0	0	0	0	0	0	-1	-1	1
e	1	1	0	0	0	0	0	0	-1	-1	-1

Table I. Input and output features used in our simulations of Marcus et al.'s (1999) experiments. Positive, negative, and zero feature values correspond to [+], [−], and unmarked feature values used in standard phonological theory, respectively.

Segment	[syllabic]	[sonorant]	[voice]	[coronal]	[continuant]	[labial]	[velar]	[nasal]	[ejective]	[high]	[tense]
p	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1
b	-1	-1	1	-1	-1	1	-1	-1	-1	-1	-1
t	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1
d	-1	-1	1	1	-1	-1	-1	-1	-1	-1	-1
n	-1	-1	1	1	-1	-1	-1	1	-1	-1	1
tʃ	-1	-1	-1	1	-1	-1	-1	-1	-1	1	-1
dʒ	-1	-1	1	1	-1	-1	-1	-1	-1	1	-1
k	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1
g	-1	-1	1	-1	-1	-1	1	-1	-1	-1	-1
f	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1
v	-1	-1	1	-1	1	1	-1	-1	-1	-1	-1
s	-1	-1	-1	1	1	-1	-1	-1	-1	-1	-1
z	-1	-1	1	1	1	-1	-1	-1	-1	-1	-1
ʃ	-1	-1	-1	1	1	-1	-1	-1	-1	1	-1
ʒ	-1	-1	1	1	1	-1	-1	-1	-1	1	-1
x	-1	-1	-1	-1	1	-1	1	-1	-1	-1	-1
ɣ	-1	-1	1	-1	1	-1	1	-1	-1	-1	-1
w	-1	1	1	-1	1	1	1	-1	-1	1	-1
j	-1	1	1	1	1	-1	-1	-1	-1	1	-1
l	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1
i	1	1	1	-1	1	-1	-1	-1	-1	1	1
o	1	1	1	-1	1	1	1	-1	-1	-1	1
e	1	1	1	-1	1	-1	-1	-1	-1	-1	1
u	1	1	1	-1	1	1	1	-1	-1	1	1
ɪ	1	1	1	-1	1	-1	-1	-1	-1	1	-1
ɔ	1	1	1	-1	1	1	1	-1	-1	-1	-1
ɛ	1	1	1	-1	1	-1	-1	-1	-1	-1	-1
ʊ	1	1	1	-1	1	1	1	-1	-1	1	-1

Table II. Input and output features used in the simulations testing generalization to novel feature values. One class of sounds is excluded from the table to save space—all segments that are -1 for [sonorant] have a counterpart that is identical, except for having a value of 1 for [ejective]. Positive feature values correspond to [+] feature values used in standard phonological theory, while negative values correspond to both [−] and unmarked feature values.