Learning reduplication with a neural network that lacks explicit variables

Brandon Prickett[1]*, Aaron Traylor*[2]*, and Joe Pater*[1]
[1] University of Massachusetts Amherst
[2] Brown University

ABSTRACT

Reduplicative linguistic patterns have been used as evidence for explicit algebraic variables in models of cognition.[1] 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 claim 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 of symbolic variables in models of cognition.

Keywords: neural networks, reduplication, symbolic computation, connectionism, generalization, phonology

[1] The authors would like to thank Max Nelson, Gaja Jarosz, Brendan O'Connor, and the members of the UMass Sound Workshop for helpful discussion and feedback. This research was funded by NSF grant BCS-1650957 to the University of Massachusetts Amherst.

1 INTRODUCTION

Identity-based patterns in language have been used as evidence for explicit algebraic variables in models of cognition (Marcus 2001; Berent 2013). 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. [linana] or [gatiti]) 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 could successfully generalize the reduplicative pattern.

Marcus *et al.* (1999) demonstrated that a simple recurrent neural network (SRN; Jordan 1986; Elman 1990) could not learn this pattern in a way that led to human-like generalization,[2] 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 generalized 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 have been made to simulate the results of the experiment without using such variables (see Shultz and Bale 2001; Endress *et al.* 2007, for a summary). The majority of these attempts have been dismissed because they either failed to produce

[2] While we choose to focus on linguistic generalizations in this paper, a considerable amount of research has also explored non-linguistic generalization (see, e.g, Doumas and Hummel 2010).

a model that discriminated between novel conforming and nonconforming items or because the model used a mechanism that was equivalent to algebraic variables (although see Alhama and Zuidema 2018 for a successful attempt, described more in Section 2). 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).

We reframe the reduplication problem within a modern context with a focus on generalization outside the training data. The remainder of the paper is structured as follows: Section 2 summarizes previous computational work on reduplication generalization, and Section 3 argues that the Sequence-to-Sequence network (Seq2Seq; Sutskever *et al.* 2014) is a straightforward architecture for sequence transduction and is a natural fit for the reduplication problem. Section 4 summarizes a series of simulations that show that a variable-free Seq2Seq network, when trained correctly, can successfully model Marcus *et al.*'s (1999) results. Section 5 then explores 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, and argues that its ability to generalize matches that which has been observed in humans. Finally, Section 6 summarizes our findings, discusses why our model was successful, suggests future work, and then concludes the paper.

BACKGROUND 2

The debate between connectionist and symbolic theories of language has often focused on the domain of morphology (for example, see Rumelhart and McClelland 1986; Pinker and 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 reduplicative processes 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).[3]

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

To model the process with a neural network, he instead used a feed-forward model that could discriminate between identical and nonidentical 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; for examples of other experimental work on reduplication, see Stemberger and Lewis, 1986 and Waksler 1999). 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 [lilina]) and those trained on ABB patterns (e.g. with sequences like [linana]). 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. [wofefe] for the ABB condition) or pattern nonconforming (e.g. [wowofe] 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.

To the best of our knowledge, the cognitive science literature lacks a formal definition for what exactly constitutes a *variable*,[4] however there is a consensus that SRNs lack any explicit variables (see, e.g., Marcus *et al.* 1999; Seidenberg and Elman 1999). Here, we use the term *explicit* to refer to a representation that has been built into a model's architecture, pretraining, or the input/output features the

[4] See Clark and Yoshinaka (2014, pp. 13–14) for some discussion of this from a formal language theoretic perspective.

model uses. While it might be the case that SRNs have the ability to capture variable-like representations in their connection weights, unless such weights were set by hand, this would not fall under our definition of an explicit variable.

Marcus *et al.* (1999) also related SRN's inability to generalize reduplication to another linguistic phenomenon – compositionality, the ability for words to combine to make novel meanings. For example, even if a person had no prior exposure to the sentence, "the bicycling iguana won the game of hop-scotch", they would be able to compose the meanings of each word to deduce the meaning of the full sentence. Additionally, even if the word "iguana" was substituted with a nonce word like "glork", humans would still be able to intuit a certain amount of meaning from the sentence. Marcus (1998) demonstrated that SRNs failed to learn human-like compositionality from linguistic data, and more modern neural networks still seem to fail at this task (Lake and Baroni 2017), unless explicit variables are built into their architecture (Korrel *et al.* 2019).

A number of attempts have been made to model the Marcus *et al.* (1999) results without the use of explicit variables. Shultz and Bale (2001) laid out diagnostics for determining whether a simulation properly demonstrates that variables are not necessary for modeling Marcus *et al.*'s (1999) results (see also Marcus 1999). The first diagnostic 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 requirement 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 reason to assume the infants in the experiment received such pretraining, this simulation failed to provide evidence for variable free models' ability to simulate Marcus *et al.*'s (1999) experiment.

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.[5]

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 are explicitly compared to one another in the model's hidden layer, they are no different from variables in regards to this criterion.

The final requirement 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 and 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 were made to model Marcus *et al.*'s (1999) results, however Shultz and Bale (2001) and Endress *et al.* (2007) argue 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 (i.e. items that followed an AAA pattern, where all three syllables in a sequence are identical). This pathological prediction by Altmann's (2002) learner will be discussed further in Section 4 where we show that our model succeeds on this new type of test item.

[5] Alhama and Zuidema (2018) also test a model without Incremental Novelty Exposure and find similar results to those presented in Section 4. We leave exploring the differences between their model and ours to future work.

OUR MODEL 3

In this section, we present the main differences between our model (a Seq2Seq network with LSTM layers) 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 (2015) and Rahman (2016). The software that we used can be downloaded at https://github.com/blprickett/Reduplication-Simulations.

We chose to focus on Seq2Seq models because of their recent success in a number of linguistic tasks (Cotterell *et al.* 2016; Kirov and Cotterell 2018; Prickett 2019; Nelson *et al.* 2020). 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 judgments correlated more with human data gathered by Albright and Hayes (2003) than any previously proposed model (although generalizing to novel verbs in a human-like way was dependent on a particular set of starting weights. See Corkery *et al.* 2019 for more on this).

Crucially for our work, 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 meets the criteria from Shultz and Bale (2001) discussed in Section 2.[6] For other recent approaches to computationally modeling reduplication, see Alhama and Zuidema (2018), Wilson (2019), Beguš (2021), Dolatian and Heinz (2020), and Haley and Wilson (2021).

Seq2Seq architecture 3.1

Seq2Seq neural networks were originally designed for machine translation and have the ability to map from one string to another, without requiring a one-to-one mapping between the strings' elements

[6] While it has become standard in machine translation for Seq2Seq models to use attention (Bahdanau *et al.* 2015), our model does not include this mechanism, since it could be considered to be an implementation of the variables that Marcus (2001) describes. See Nelson *et al.* (2020) for a discussion of how attention can help neural networks learn reduplicative patterns.

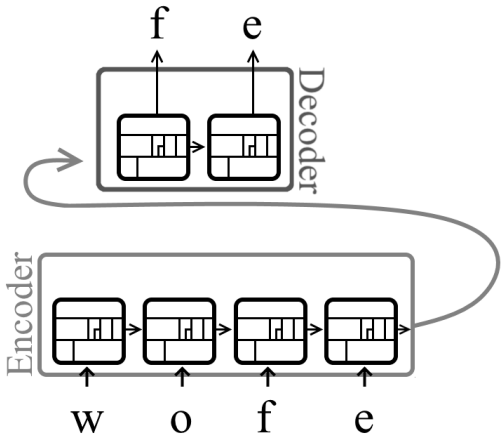

Figure 1:
Illustration of Seq2Seq architecture modeling one of the stimuli (represented as a mapping from the first two syllables to the third syllable) in Marcus *et al.*'s (1999) experiments. Each transcribed sound represents a single timestep

(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, with the encoder processing the input string one element at a time and the decoder transforming that processed data into an output string that it unpacks through time. Often these elements that make up the input and output sequences are referred to as "timesteps". In our simulations, each timestep represents a single phonological segment as either a vector of arbitrary features or a vector of phonetically motivated features adapted from the phonological literature.

Figure 1 shows an illustration of the Seq2Seq architecture that resembles the mappings we use in the simulations described in Section 4.1. Here, the encoder passes through the entire input (i.e. the first two syllables) before transferring information (in the form of hidden layer activations) 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 long and the output being two.

At each timestep in the input, information is passed forward through the hidden layers of the encoder (represented in the figure by the black boxes within the encoder). Additionally, information is

passed *across* timesteps through the model's recurrent connections (represented by the black, rightward pointing arrows in the figure). The final recurrent connection in the encoder (represented by the gray arrow) passes this processed information to the decoder, which unpacks it timestep-by-timestep in the output. In all of the simulations discussed in this paper, the encoder is unidirectional, meaning that it passes through the input string once, from left to right.

Long Short-Term Memory (LSTM) 3.2

In all of the simulations presented here, our model uses LSTM hidden layers (Hochreiter *et al.* 2001). These are a kind of recurrent neural network layer which enhances a model's ability to store information over several 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 layers can also provide models with added representational power (Levy *et al.* 2018).

LSTM performs both of these by using cell states: bundles of interacting layers that can learn which information is important for the model to keep track of in the long term, and which information it can forget. This means that during training, the network is not only learning how to predict the output from the input at a given point in time, but also which information at that timestep will help it to predict the output in the future. Crucially, nothing in LSTMs explicitly implements an algebraic variable. While the use of LSTM layers likely has some effect on our model's predictions, we do not expect it to be the primary factor affecting the network's generalization and leave the question of how crucial this mechanism is to future work.

Dropout 3.3

Dropout is a regularization method that helps neural networks generalize correctly to items outside of their training data (Srivastava *et al.* 2014). When using dropout, a hyperparameter is chosen between 0 and 1 that represents the probability that any given unit in the network is "dropped out" during training (i.e. all of its incoming/outgoing weights are temporarily set to 0). The set of units that are dropped out

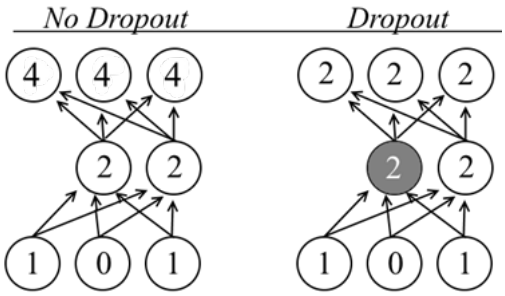

Figure 2:
A simple feed-forward network, with and without dropout. Each circle is a unit and each arrow is a connection. Dropped out units are in gray. Each unit's output (before dropout) is denoted by the number inside of it. All connections have a weight of 1 and all activation functions are $f(x)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

is resampled at each weight update during learning, forcing the model to find a solution that does not depend too heavily on any single unit.

This is illustrated for a simple feed-forward network on the right side of Figure 2. 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. For the simulations presented here, dropout was applied with equal probability to all layers of the network.

4 MODELING MARCUS *ET AL.* (1999)

This section presents simulations of the three experiments described in Marcus *et al.* (1999). In addition to directly simulating these three experiments, Section 4.1 explores the impact of linguistic structure in the model's pretraining, and Section 4.2 simulates a partial replication of the original Marcus *et al.* (1999) experiment performed by Endress *et al.* (2007).

4.1 *Experiments 1 and 2*

In their first two experiments, Marcus *et al.* (1999) trained infants on ABB and ABA patterns (e.g. [wofefe] and [wofewo], respectively) and then measured the infants' listening times to determine whether they generalized the patterns to words containing novel segments. To simulate this, we trained our model to predict the third syllable in each

experimental item, based on the first two.[7] 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 (1999) experiments that they describe, are given in the Supplementary Materials. Both the encoder and decoder each had 4 LSTM layers with 11 units in each layer.

The model was trained using RMSProp (Tieleman and Hinton 2012), a gradual, error-based algorithm, with the default hyperparameter values used in Keras (Chollet 2015). The probability of dropout was .85 (chosen after a small amount of pilot testing before running our final simulations) and the loss that the model was trained to minimize was mean squared error (MSE). MSE was calculated by going through each feature in the model's predicted output, squaring the difference between the predicted value of this feature and the correct value, and averaging across all of these squared differences.

In addition to being trained on the same items as Marcus *et al.*'s (1999) subjects, given in Table 1, 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 the pretraining used by 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

[7] Note that this mapping is much simpler than some kinds of reduplication present in natural language (see, e.g., Dolatian and Heinz 2020, for more on this) and should be trivially easy for a neural network to learn. However, since Marcus *et al.* (1999) and the current study are primarily interested in generalization, the formal complexity and learnability of the patterns we look at is irrelevant.

Table 1: Training data used in our simulations of the first two experiments in Marcus *et al.* (1999). For the phonological features used to represent each sound, see the Supplementary Materials

Experiment	Condition	Stimuli
1	ABA	[gatiga], [ganaga], [gagiga], [galaga], [litili], [ligili], [lilali], [nigini], [ninani], [nilani], [talata], [tatita], [linali], [nitini], [tanata], [tagita]
1	ABB	[tigaga], [nagaga], [gigaga], [lagaga], [tilili], [gilili], [lalili], [ginini], [nanini], [lanini], [latata], [titata], [nalili], [tinini], [natata], [gitata]
2	ABA	[ledile], [lejele], [lelile], [lewele], [widiwi], [wijewi], [wiliwi], [wiwewi], [jidiji], [jijeji], [jiliji], [jiweji], [dedide], [dejede], [delide], [dewede]
2	ABB	[dilele], [jelele], [lilele], [welele], [diwiwi], [jewiwi], [liwiwi], [wewiwi], [dijiji], [jejiji], [lijiji], [wejiji], [didede], [jedede], [lidede], [wedede]

of two randomly sampled syllables that mapped to another randomly chosen syllable.

After being trained on 1000 of these randomly produced data for 1000 epochs (i.e. full passes through the data) with batches of size 50 (i.e. the model made weight updates based on the average error on 50 data points), the model's decoder weights were set back to their original values (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 four test items used by Marcus *et al.* (1999): [wofefe], [dekoko], [wofewo], and [dekode] for Experiment 1 and [bapopo], [kogaga], [bapoba], [kogako] for Experiment 2. Testing involved feeding the model a set of prespecified input values and comparing the model's resulting output values to the correct outputs (as mentioned above, this comparison is reported using MSE). We used

Table 2: Results from our simulations and the corresponding experiments in Marcus *et al.* (1999)

	MSE				*Listening Time*			
	Conf.	Nonconf.	$t(99)$	p	Conf.	Nonconf.	F(14)	p
Exp. 1	.49	.52	−2.8	<.01*	6.3	9.0	25.7	<.01*
Exp. 2	.67	.68	−3.3	<.01*	5.6	7.35	25.6	<.01*

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[8] (50 per condition, per experiment) are given in Table 2, along with the results reported by Marcus *et al.*'s (1999) 32 subjects (8 per condition, per experiment). All MSE values are rounded to the nearest hundredth and averaged across runs.

The results in Table 2 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.[9] This means that the nonconforming stimuli were predicted more poorly by the model, meeting the final diagnostic laid out by Shultz and Bale (2001) for knowing whether a simulation successfully captures the infants' behavior without explicit variables.

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, for a number of reasons. First of all, the comparison we make above assumes a linking hypothesis in which each run of the Seq2Seq network is equivalent to a single infant in the experiment. However, it is not obvious that this is the correct

[8] 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.

[9] 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, with differences in average MSEs between conforming and nonconforming items of 0.034 and 0.019, respectively.

assumption. For example, one could imagine combining the results from several separate runs to simulate a single human's behavior in the experiment (for example, by averaging their MSE on the test stimuli). This kind of "ensemble" technique, where predictions are combined from multiple models with the same training data, is common in machine learning (Kuncheva 2014) and would reduce the variability in our results (thus increasing the effect size). While it is difficult to determine what linking hypothesis is the most realistic, ensemble learning is an example of how the variable-free model we use here could have an effect size comparable to that of the infants.

Another way that we could reduce the variability and increase the effect size in our results is by reducing the range in which the model's initial weights can vary. Currently, each connection's weight was randomly chosen at the start of each run, which is why each repetition of the simulation got different results, despite getting similar pretraining and identical training data. However, the variability present in these initial weights was due to the default settings in the software we were using, rather than any principled measurement based on actual variability in the brains of newborns. It could be that infants have a relatively low level of variability in their initial state of learning – and if we replicated this in our simulations it could increase our effect size considerably, since we could choose a set of starting weights that led to high levels of generalization and low amounts of variance across runs (for work that pursues this possibility in the context of other phonological patterns, see McCoy *et al.* 2020).

Finally, since the infants would have been exposed to repetition in language prior to their participation in the experiment, their learning could have been aided by this previous linguistic experience. Examples of reduplication are common in both infant-directed speech (Ferguson 1964; Mazuka *et al.* 2008) and adult English (Nevins and Vaux 2003; Ghomeshi *et al.* 2004; Štekauer *et al.* 2012). For example, many of the words directed toward infants (such as "mama" and "choochoo") contain repetition that could be considered an ABB pattern (since two adjacent syllables repeat). Similarly, *Shm* Reduplication (e.g. "pizza-shmizza"; Nevins and Vaux 2003) could be represented as an ABA pattern, with the B representing the [ʃm] sequence and the A's representing the copied material. Mazuka *et al.* (2008) estimate that as much as

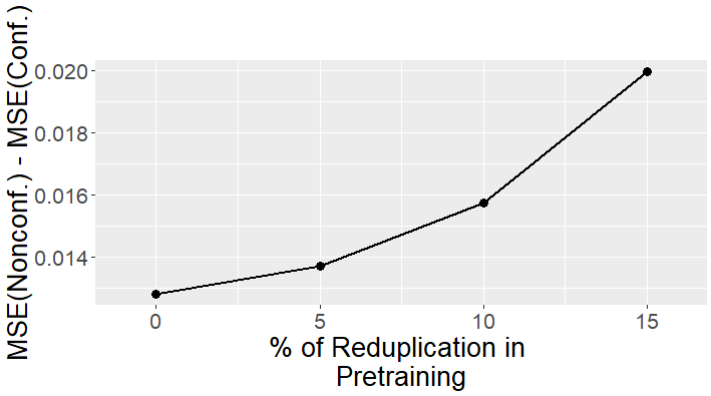

Figure 3:
The effect of reduplication in pretraining on the effect size of the results. Each datapoint represents the average difference between the MSE of conforming and nonconforming items, over 100 repetitions. Half of the simulations were in the ABB experiment condition and half were in the ABA condition

65% of the word types in infant-directed speech could contain some kind of repetition, based on self reporting from Japanese mothers.

We tested the hypothesis that infants might be aided by native language reduplication by running another set of simulations in which we added ABB and ABA conforming words to the model's pretraining. We varied the percentage of the pretraining that contained these reduplicative words to see if more reduplication in pretraining changed the effect size when simulating the experiments. Additionally, we added a feature to represent the semantic information that would be associated with this repetition. In pretraining, this semantic feature was always −1 when words followed an ABA pattern and 1 whenever words were ABB. When simulating the experiment, this feature was always 0 (to represent the lack of meaning associated with the experimental stimuli). All other hyperparameters were the same as the simulations described above, and the results from them are shown in Figure 3.

Figure 3 demonstrates that the more repeating items that were added into the model's pretraining, the larger its effect size became when simulating the experiment. While the effect size of the model does not reach the same levels as the infants in Marcus *et al.*'s (1999) study, this demonstrates that adding structure into the model's pretraining does have the potential to increase effect size. Since the infants in the study were exposed to much more linguistic structure than just reduplication, the benefit they received from their prior experience with English could have had an even larger influence on their ability to generalize in an experimental setting.

Table 3: Training data used in our simulations of the third experiment in Marcus *et al.* (1999). For the phonological features used to represent each sound, see the Supplementary Materials

Experiment	Condition	Stimuli
3	AAB	[leledi], [leleje], [lelеli], [lelewe], [wiwidi], [wiwije], [wiwili], [wiwiwe], [jijidi], [jijije], [jijili], [jijiwe], [dededi], [dedeje], [dedeli], [dedewe]
3	ABB	[dilele], [jelele], [lilele], [welele], [diwiwi], [jewiwi], [liwiwi], [wewiwi], [dijiji], [jejiji], [lijiji], [wejiji], [didede], [jedede], [lidede], [wedede]

4.2 *Experiment 3*

Marcus *et al.*'s (1999) third experiment required a different set-up than our previous simulations. As shown in Table 3, this experiment replaced the ABA pattern with AAB, exposing infants to either this or ABB words in training, depending on the condition they were assigned.

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 ABB. 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 that the empty syllable was supposed to include (see Devlin *et al.* 2019, for a similar approach in natural language processing). This is illustrated in Figure 4.

Since the second syllable is predictable in both the AAB and ABB conditions, given the other two syllables, this allowed us to test the model on a mapping that was relevant to the design of Experiment 3. While it is unlikely that infants were performing this exact task, framing the problem in this way allows us to work within the constraints of the Seq2Seq network (i.e. that all tasks are a string to string mapping)

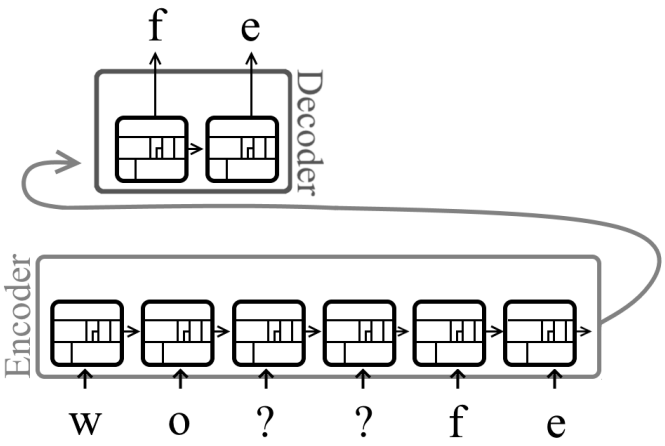

Figure 4:
Illustration of an ABB mapping in Experiment 3's simulations. The "?" symbols represent empty segments

and still ensure that, like the infants, the model is not just learning to attend to changes across syllable boundaries.

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 then trained on a data set similar to those in the previous section. The test phase was also similar to the other experiments, with the model being tested on the words [bapopo], [kogaga], [babapo], and [kokoga]. The results on these test items, averaged over 20 simulations (10 in each condition) are shown in Table 4.

MSE				*Listening Time*			
Conf.	Nonconf.	$t(19)$	p	Conf.	Nonconf.	F(14)	p
.56	.57	−2.3	.01635*	6.4	8.5	40.3	<.001*

Table 4:
Results for the Experiment 3 simulation, compared to Marcus *et al.*'s (1999)

The Experiment 3 simulations also included an additional kind of test item. This 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 explore a prediction made by Altmann's (2002) model. That model correctly predicted a preference for conforming stimuli over

Table 5: Results on Endress *et al.*'s (2007) conforming and nonconforming stimuli

MSE			
Conf.	Nonconf. (AAA)	$t(19)$	p
.56	.57	−2.22	.01933*

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 [bababa], 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 participants' 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 also tested it on the Endress *et al.* (2007: Appendix A) test items: [bababa] and [kokoko]. The results, averaged over 20 simulations (10 in each condition), are given in Table 5.

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) partial 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 ($t[19] = -2.22$, $p = .01933$).

5 EXPLORING THE MODEL'S SCOPE OF GENERALIZATION

In Section 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 not present in the words that infants were trained on), however, all of the segments in the experiment

	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 6:
Example of generalization to a novel syllable.
Gray cells represent training data,
bolded item indicates the crucial testing item

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, which entails that the model never needed to generalize reduplication to completely novel phonemes. This also means that it is impossible to know, based on those results, whether the model learned an algebraic function like $\alpha\beta_1 \rightarrow \beta_2$, or whether it learned a less general pattern like "if feature F is 1 in the third sound in the input, feature F' should be 1 in the first sound of the output".

To better understand the mappings being learned by the Seq2Seq network, we structured the simulations in this section to map a single syllable (e.g. [ba]) to two copies of itself (e.g. [baba]).[10] 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 identity-based patterns. We summarize the three scopes here, and then in Section 5.1–5.3, we explain the series of simulations we ran to determine which scope best describes our model's performance.

The simplest form of generalization that Berent (2013) discussed is to novel words (which in this context is equivalent to generalization to novel syllables, since the network is blind to the difference between these levels of representation). This is illustrated for a reduplicative pattern in Table 6, with the gray cells representing the input syllables seen in the training data and the bolded syllable being the input for a test item withheld from training.

[10] This also resembles natural language reduplication more closely than the Marcus *et al.* (1999) pattern does. For an example, see reduplication in the language Karao, which doubles the stem of a word to change the number of some verbs: [man**ba**kal] "fight each other, 2 people" → [man**baba**kal] "fight each other, >2 people" (Štekauer *et al.* 2012).

Table 7: Example of generalization to a novel segment. Gray cells represent training data, bolded items indicate crucial testing item

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

If a model correctly predicts the mapping [da]→[dada] after being trained on data that does not include the input [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 Section 4, we represent segments as vectors of phonological features. When testing this scope, we trained the model on every relevant value for each feature, but not on all of the possible feature value combinations. This is demonstrated in Table 7, using the same shading scheme that was described above.

In the example in Table 7, the model is trained on syllables containing [p], [b], and [t], with [d] remaining outside of its training data. 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 generalizes at this level, it has acquired a broader generalization that might reference specific feature values. For example, it may have learned 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 full identity-based function. To show this, a model would need to demonstrate

	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 8:
Example of generalization to a novel feature value. Gray cells represent training data, bolded item indicates the crucial testing item

its ability to generalize to novel feature values, which Berent (2013) calls "across the board" generalization and Marcus (1998) describes as "outside of the training space".[11] This is demonstrated in Table 8, where the learner is only trained on oral consonants (i.e. sounds made without nasal resonance) and then tested on the nasal consonant [n].

In the example from Table 8, 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 value 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 syllable.

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 (Section 5.1), one in which the model was tested on a syllable made with a segment that it had not received in training (Section 5.2), and one in which the model was tested on a syllable with a novel segment containing a feature value that had not been presented in the training data (Section 5.3). None

[11] Note that all scopes of generalization talked about so far can be thought of as being "outside the training space", but Marcus and colleagues often use this term to specifically refer to generalization to novel feature values. By "feature" here we mean the most atomic level of a model's representation. For our model, this is the level of phonological features, following standard linguistic theory (see, e.g., Chomsky and Halle 1968).

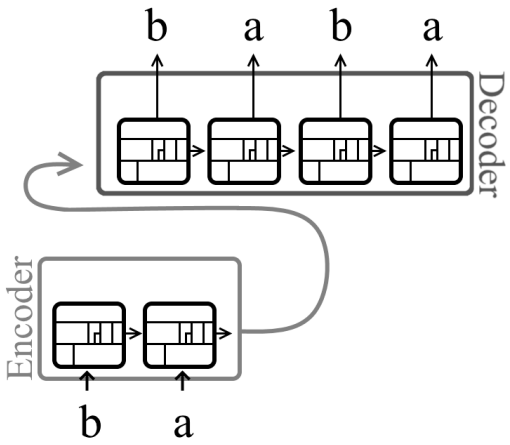

Figure 5:
Illustration of mappings
in this section's simulations

of the simulations described here used a pretraining phase like those in Section 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, unless otherwise noted. Input 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 these two categories 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 mappings 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 5.

The models were trained for 1000 epochs, with batches that included all of the training 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 Section 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), which performs a single forward pass through the network. Since the model is not probabilistic, these predictions do not vary given the same input and set of connection weights. These 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.

Generalization to novel syllables 5.1

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 that 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%).

Generalization to novel segments 5.2

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

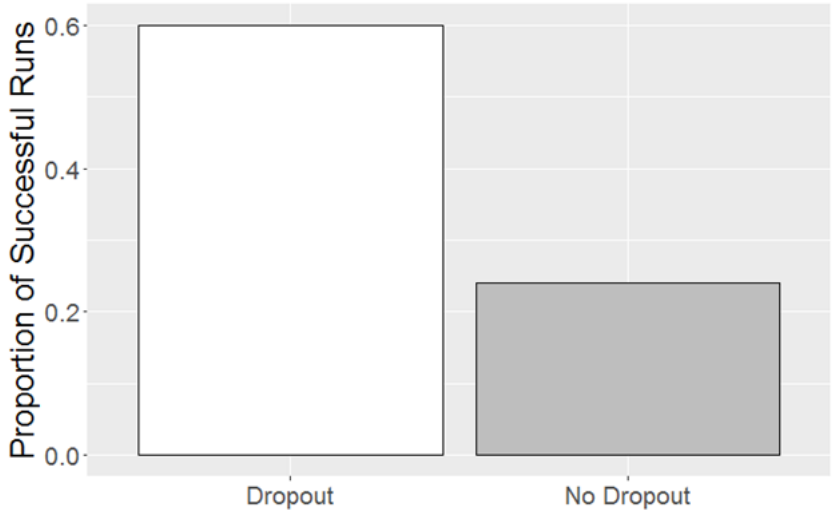

Figure 6: Difference between dropout with probabilities of .75 and 0 in generalization to novel segments

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 6.

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 individual features, rather than completely abstract algebraic functions like $\alpha \rightarrow \alpha\alpha$. In this condition, the

inventory was designed by hand and always contained 43 segments, in order to more easily withhold a single feature value. The feature vectors that represented these segments are given in the Supplementary Materials. 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 achieved 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 whether they are regularized with dropout, cannot generalize to novel feature values.[12]

5.4 *Which scope of generalization is observed in human language learning?*

In this section, we argue that the generalization observed in our Seq2Seq simulations matches the generalization demonstrated in past experiments involving humans. As we'll discuss, the ability of humans to generalize identity-based patterns to novel words and segments is well documented and uncontroversial, but we find that the evidence for humans generalizing to novel feature values is weak.

When discussing generalization of reduplicative patterns, Berent (2013) used Hebrew speakers' judgments regarding an AAB 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 consonants are not allowed to match the pattern AAB, where the A's represent a repetition of the same consonant). 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.

[12] One reason why you might expect this behavior is that novel feature values represent a particularly strong violation of the "independent and identically distributed" assumption (see Le Boudec 2011, for an introduction) often made in statistical learning.

Generalization to novel words/syllables was demonstrated by Berent and Shimron (1997) in an experiment that asked Hebrew-speaking participants to rate nonce words. These words were made up of segments that were attested in Hebrew, such as [s] and [m], making them equivalent to the novel syllables that we tested our model on in Section 5.1. Speakers in this experiment rated words with s-s-m stems (like *[sisem]) 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).

Generalization to novel segments by Hebrew speakers was shown in Berent *et al.* (2002), corresponding to the scope of generalization that the network with dropout achieved in Section 5.2. 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 *et al.* (2002) showed that speakers can generalize the *AAB pattern to the segment [θ], which they claimed represented generalization to the novel feature value [wide]. However, [wide] is not used in any standard phonological feature theory (e.g. Chomsky and 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). This is illustrated in Table 9.

Berent *et al.* (2002) present a number of arguments in favor of using the feature [wide], rather than a more standard phonological representation. First, they argue that since [wide] is a more phonetically invariant feature value than representations like [+anterior] and [−strident], that it is more likely to be psychologically real (see also Gafos 1999). However, it is unclear whether phonological features *should* have invariant phonetic correlates (see Hamann 2010,

	[anterior]	[continuant]	[strident]
t	+	−	
ʃ	−	+	+
f		+	−
θ	+	+	−

Table 9:
Demonstration that [θ] does not represent any novel feature values for Hebrew speakers when a standard set of features is used. Gray cells represent the crucial feature values needed to describe [θ] that are present in native Hebrew sounds

sec. 2.1 for some discussion of this), since the process that the mind uses to map phonetic information to phonological features is an open question.

Their second piece of evidence was that Hebrew speakers map [θ] to [t] when borrowing non-Hebrew words, and that this must be the result of a representational difference between it and other novel sounds that are borrowed faithfully into the language. However, it has been widely observed that interdental sounds like [θ] are more likely to be mapped incorrectly than other phonemes when words containing them are borrowed into a language (see, e.g., Rau *et al.* 2009; Hanulikova and Weber 2010). This is likely due to phonetic difficulty, since children acquiring English as their first language are more likely to make production (Moskowitz 1975) and perception (Skeel 1969) errors when dealing with interdental sounds than other kinds of phonemes.

Another experiment claiming to demonstrate generalization to novel feature values is Berent *et al.* (2014). In this paper, the authors claim to observe generalization of a reduplicative pattern in American Sign Language to novel signs made up of novel feature values. However, they are using the word "feature" differently than we do here. While they define features as 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 (for example, the position of individual fingers). Since their participants would have had prior linguistic and non-linguistic experience with visual stimuli that involved hands in a variety of positions (analagous to the pretraining we used in Section 4), these would not be truly novel feature values for them. Berent *et al.* (2016) also used signed language to test whether humans could generalize to novel feature values. Specifically, they showed that na-

tive speakers of auditory languages seemed to generalize reduplicative patterns from their L1 to signed nonce words. However, this study also used visual stimuli that could easily be represented using features that participants were already exposed to in non-linguistic contexts (i.e. the different positions of the parts of a hand). Furthermore, since the participants in the experiment were not experienced speakers of a signed language, they could have been mapping the signs to auditory representations in their mind, which would mean that they were not generalizing the reduplicative patterns to novel features at all.

To our knowledge, no experiment has conclusively tested humans' ability to generalize to novel feature values. Such an experiment would be difficult, since children stop reliably perceiving most novel feature contrasts at a relatively young age (see, e.g., Werker and Tees 1983). Because of this, we conclude that our model generalizes in a way that captures the scopes observed thus far in human behavior: generalization to novel syllables and generalization to novel segments.

6 DISCUSSION

6.1 *Summary of results*

In Section 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. Results from these simulations are summarized in Figure 7.

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: Appendix A).

Next, we probed our model further in Section 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 in Figure 8.

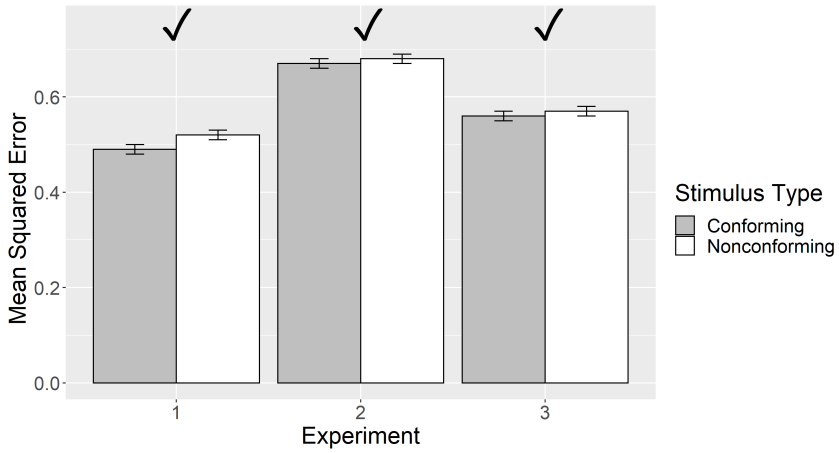

Figure 7: Results from our simulations of the three experiments described in Marcus *et al.* (1999). Error bars show standard error of the mean, check symbols indicate successful simulations of the behavior observed in each experiment

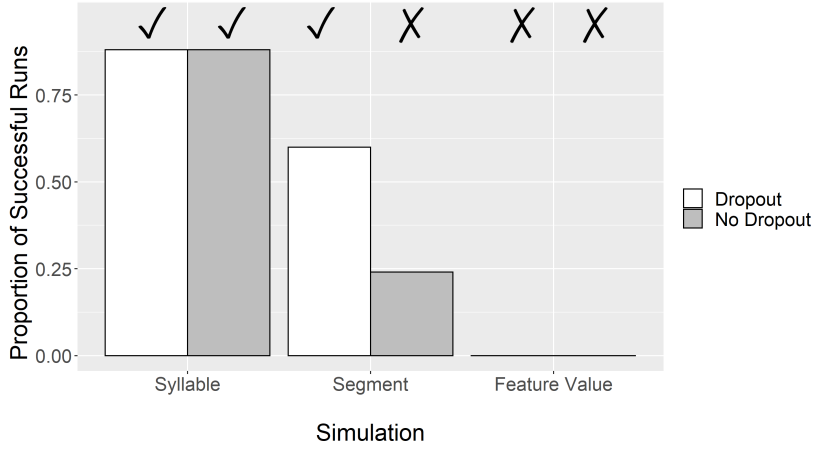

Figure 8: Summary of results for each dropout condition and scope of generalization. Checks and 'X' symbols indicate which conditions the model reliably succeeded and failed in, respectively

The findings from this series of simulations showed that even without dropout, a Seq2Seq model is not simply memorizing mappings for each individual datum, since it was able to generalize reduplication to novel syllables. We also showed that the model, when using dropout in training, can reliably generalize reduplication to novel segments. However, generalization to novel feature values was never achieved, regardless of whether or not dropout was used.

6.2 *Why can the Seq2Seq model learn generalizable reduplication?*

Why neural networks generalize in the way that they do is still an open question (see, e.g., Valle-Perez *et al.* 2018). However, the results presented in Section 5 shed some light on why our model succeeded in capturing the infant behavior reported by Marcus *et al.* (1999), while past neural networks failed (for similar work on probing neural networks using generalization tasks, see, e.g., Linzen *et al.* 2016; McCoy *et al.* 2018). First of all, we found that the network could never generalize to novel feature values. This explains why past models that were given no pretraining could not capture the infant generalization – since the pretraining exposed our model to all of the feature values present in both the training and testing phase of the experiment.

Additionally, we found that generalization to novel segments only occurred reliably for our model when it used dropout (Srivastava *et al.* 2014), a standard regularization technique in machine learning. This also explains the failure of past models, since (to our knowledge) dropout has not been used in past attempts to simulate the experiment (although, see Alhama and Zuidema 2018, for the sucessful application of a related mechanism).

It remains an open question whether other forms of regularization (such as an L2 prior) would be as successful at this task as dropout was. One hypothesis for why dropout worked is that it caused certain training data to be indistinguishable from crucial testing data. For example, if the training set included the inputs [pa] and [da], but [ta] was withheld, a model without dropout would not generalize to the novel item because it was never trained on reduplicating [t]. However, if dropout is applied, then in a subset of epochs, the unit activations distinguishing [t] from [d] would no longer be available to the

model. This would allow it to learn how to reduplicate a syllable that is ambiguous between [ta] and [da]. While this would not allow the model to generalize to novel feature values that were never activated in training, it 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. Testing these other methods is an important avenue that future research should explore.

6.3 *Future work*

There are a number of other opportunities that present themselves for future work. For example, running experiments on humans that test for generalization of reduplicative patterns to truly novel feature values (if such a test is possible) would be beneficial, since it would help shed more light on what scope of generalization computational models need to achieve.

Probing the Seq2Seq model further to better understand the representations it learns when acquiring reduplication is another important direction for future research to investigate. Our results suggest that when dropout is used, the model is likely learning a feature-based representation, but understanding which parts of the model's architecture are responsible for this is still an open question. Methods exist for probing networks in this way (see, e.g., Beguš 2021; Dankers *et al.* 2021) and could help shed light on what exactly is neccessary for a model to capture the results from Marcus *et al.* (1999).

Another area future research should pursue is the relationship between formal descriptions of reduplication (e.g. Clark and Yoshinaka 2014; Dolatian and Heinz 2020; Wang 2021) and the results discussed here. While both experimental (Moreton *et al.* 2021) and computational (Nelson *et al.* 2020) work has touched on the formal complexity of reduplication, there is still much work to be done to bridge the kind of modeling done here with models like finite state automata that are often used to more precisely describe the learnability of patterns.

The learning biases inherent to the Seq2Seq model should also be explored. For example, Endress et al. (2007) and Gallagher (2013) both found that identity-based patterns were easier for humans to

learn than more arbitrary ones, and concluded that explicit variables were necessary to model this behavior. Testing to see whether Seq2Seq networks with dropout show a similar bias for identity-based patterns could be another way of testing whether variables are needed in models of cognition.

Additionally, the question of compositionality should be revisited, given our findings on reduplication. If neural networks' ability to model these two phenomena is related, as Marcus *et al.* (1999) suggested, then given the right pretraining, a Seq2Seq network with dropout should be able to learn compositional linguistic patterns. Capturing compositionality may require testing novel kinds of featural representations, since our results suggest that novel feature values in the input or output will always be impossible for the model to generalize to (see Lake and Baroni 2017, sec. 5, for a similar suggestion).

6.4 *Conclusions*

In the past, it has been claimed that it is impossible for variable-free neural networks to generalize reduplicative patterns in a human-like way (Marcus *et al.* 1999; Marcus 2001; Berent 2013). Here, we presented results showing that a network with no variables, that has 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 simulation of the experiment, our results challenge the claim that simulating these results is only possible with a symbolic model of cognition.

We also probed our model's 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.

REFERENCES

Adam ALBRIGHT and Bruce HAYES (2003), Rules vs. analogy in English past tenses: A computational/experimental study, *Cognition*, 90(2):119–161.

Raquel G. ALHAMA and Willem ZUIDEMA (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.

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

Dzmitry BAHDANAU, Kyunghyun CHO, and Yoshua BENGIO (2015), Neural machine translation by jointly learning to align and translate, in Yoshua BENGIO and Yann LECUN, editors, *3rd International Conference on Learning Representations, Conference Track Proceedings*.

Gašper BEGUŠ (2021), Identity-based patterns in deep Convolutional Networks: Generative Adversarial Phonology and reduplication, *Transactions of the Association for Computational Linguistics*, 9:1180–1196.

Yoshua BENGIO, Patrice SIMARD, and Paolo FRASCONI (1994), Learning long-term dependencies with gradient descent is difficult, *IEEE Transactions on Neural Networks*, 5(2):157–166.

Iris BERENT (2013), The phonological mind, *Trends in Cognitive Sciences*, 17(7):319–327.

Iris BERENT, Outi BAT-EL, Diane BRENTARI, Amanda DUPUIS, and Vered VAKNIN-NUSBAUM (2016), The double identity of linguistic doubling, *Proceedings of the National Academy of Sciences*, 113(48):13702–13707.

Iris BERENT, Amanda DUPUIS, and Diane BRENTARI (2014), Phonological reduplication in sign language: Rules rule, *Frontiers in Psychology*, 5(560):1–15.

Iris BERENT, Gary MARCUS, Joseph SHIMRON, and Adamantios I. GAFOS (2002), The scope of linguistic generalizations: Evidence from Hebrew word formation, *Cognition*, 83(2):113–139.

Iris BERENT and Joseph SHIMRON (1997), The representation of Hebrew words: Evidence from the obligatory contour principle, *Cognition*, 64(1):39–72.

François CHOLLET (2015), Keras, `https://github.com/keras-team/keras`.

Noam CHOMSKY and Morris HALLE (1968), *The sound pattern of English*, Harper & Row.

Morten H. CHRISTIANSEN and Suzanne L. CURTIN (1999), The power of statistical learning: No need for algebraic rules, in Martin HAHN and Scott C. STONESS, editors, *Proceedings of the 21st Annual Conference of the Cognitive Science Society*, pp. 114–119, Routledge.

Alexander Clark and Ryo Yoshinaka (2014), Distributional learning of parallel multiple context-free grammars, *Machine Learning*, 96(1–2):5–31.

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

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater (2019), Are we there yet? Encoder-decoder neural networks as cognitive models of English past tense inflection, in Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3868–3877.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden (2016), The SIGMORPHON 2016 shared task—morphological reinflection, in Micha Elsner and Sandra Kuebler, editors, *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pp. 10–22.

Verna Dankers, Anna Langedijk, Kate McCurdy, Adina Williams, and Dieuwke Hupkes (2021), Generalising to German plural noun classes, from the perspective of a Recurrent Neural Network, *Conference on Computational Natural Language Learning*, https://aclanthology.org/2021.conll-1.8.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019), BERT: Pre-training of deep bidirectional transformers for language understanding, in Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, https://aclanthology.org/N19-1423.

Hossep Dolatian and Jeffrey Heinz (2020), Computing and classifying reduplication with 2-way finite-state transducers, *Journal of Language Modelling*, 8(1):179–250.

Leonidas Doumas and John E. Hummel (2010), A computational account of the development of the generalization of shape information, *Cognitive Science*, 34(4):698–712.

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

Ansgar D. Endress, Ghislaine Dehaene-Lambertz, and Jacques Mehler (2007), Perceptual constraints and the learnability of simple grammars, *Cognition*, 105(3):577–614.

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

Adamantios I. Gafos (1999), *The articulatory basis of locality in phonology*, Taylor & Francis.

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

Jila GHOMESHI, Ray JACKENDOFF, Nicole ROSEN, and Kevin RUSSELL (2004), Contrastive Focus Reduplication in English (The Salad-Salad Paper), *Natural Language & Linguistic Theory*, 22(2):307–357, ISSN 0167-806X, https://www.jstor.org/stable/4048061.

Coleman HALEY and Colin WILSON (2021), Deep neural networks easily learn unnatural infixation and reduplication patterns, *Proceedings of the Society for Computation in Linguistics (SCiL)*, pp. 427–433.

Silke HAMANN (2010), Phonetics-phonology interface, in Nancy C. KULA, Bert BOTMA, and Kuniya NASUKAWA, editors, *The Bloomsbury Companion to Phonology*, Bloomsbury Companions, Bloomsbury.

Adriana HANULIKOVA and Andrea WEBER (2010), Production of English interdental fricatives by Dutch, German, and English speakers, in Magdalena WREMBEL, Malgorzata KUL, and Katarzyna DZIUBALSKA-KOLACZYK, editors, *New Sounds 2010: Sixth International Symposium on the Acquisition of Second Language Speech*, pp. 173–178, Peter Lang Verlag.

Bruce HAYES (2011), *Introductory phonology*, John Wiley & Sons.

Sepp HOCHREITER, Yoshua BENGIO, Paolo FRASCONI, and Jürgen SCHMIDHUBER (2001), *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*, A field guide to dynamical recurrent neural networks. IEEE Press.

Michael I. JORDAN (1986), Serial order: A parallel distributed processing approach, Technical report, University of California, San Diego.

Christo KIROV and Ryan COTTERELL (2018), Recurrent Neural Networks in linguistic theory: Revisiting Pinker & Prince (1988) and the past tense debate, *Transactions of the Association for Computational Linguistics*, 6:651–665.

Kris KORREL, Dieuwke HUPKES, Verna DANKERS, and Elia BRUNI (2019), Transcoding compositionally: Using attention to find more generalizable solutions, in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 1–11, Association for Computational Linguistics, doi:10.18653/v1/W19-4801.

Ludmila I. KUNCHEVA (2014), *Combining pattern classifiers: methods and algorithms*, John Wiley & Sons.

Brenden M. LAKE and Marco BARONI (2017), Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks, in Jennifer DY and Andreas KRAUSE, editors, *Proceedings of the 35th International Conference on Machine Learning*.

Jean-Yves Le Boudec (2011), *Performance evaluation of computer and communication systems*, Epfl Press.

Omer Levy, Kenton Lee, Nicholas Fitzgerald, and Luke Zettlemoyer (2018), Long Short-Term Memory as a dynamically computed element-wise weighted sum, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 732–739.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg (2016), Assessing the ability of LSTMs to learn syntax-sensitive dependencies, *Transactions of the Association for Computational Linguistics*, 4:521–535.

Gary Marcus (1998), Rethinking eliminative connectionism, *Cognitive Psychology*, 37(3):243–282.

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

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

Gary Marcus, Sugumaran Vijayan, S. Bandi Rao, and Peter M. Vishton (1999), Rule learning by seven-month-old infants, *Science*, 283(5398):77–80.

Reiko Mazuka, Tadahisa Kondo, and Akiko Hayashi (2008), Japanese mothers' use of specialized vocabulary in infant-directed speech: infant-directed vocabulary in Japanese, in *The origins of language*, pp. 39–58, Springer.

R. Thomas McCoy, Erin Grant, Paul Smolensky, Thomas L Griffiths, and Tal Linzen (2020), Universal linguistic inductive biases via meta-learning, *Proceedings of the 42nd Annual Conference of the Cognitive Science Society*.

Richard Thomas McCoy, Robert Frank, and Tal Linzen (2018), Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks, in Chuck Kalish, Martina Rau, Jerry Zhu, and Timothy Rogers, editors, *Proceedings of CogSci 2018*, pp. 2096–2101.

Elliott Moreton, Brandon Prickett, Katya Pertsova, Josh Fennell, Joe Pater, and Lisa Sanders (2021), Learning repetition, but not syllable reversal, in Ryan Bennett, Richard Bibbs, Mykel L. Brinkerhoff, Max J. Kaplan, Stephanie Rich, Amanda Rysling, Nicholas Van Handel, and Maya Wax Cavallaro, editors, *Proceedings of the Annual Meetings on Phonology*.

Breyne Arlene Moskowitz (1975), The acquisition of fricatives: A study in phonetics and phonology, *Journal of Phonetics*, 3(3):141–150.

Max Nelson, Hossep Dolatian, Jonathan Rawski, and Brandon Prickett (2020), Probing RNN Encoder-Decoder generalization of subregular functions using reduplication, *Proceedings of the Society for Computation in Linguistics (SCiL)*, pp. 31–42.

Andrew Nevins and Bert Vaux (2003), Metalinguistic, shmetalinguistic: The phonology of shmreduplication, in J. Cihlar, A. Franklin, D. Kaiser, and

J. KIMBARA, editors, *Proceedings from the 39th Annual Meeting of the Chicago Linguistic Society*, pp. 702–721, Chicago Linguistic Society.

Steven PINKER and Alan PRINCE (1988), On language and connectionism: Analysis of a parallel distributed processing model of language acquisition, *Cognition*, 28(1):73–193.

Brandon PRICKETT (2019), Learning biases in opaque interactions, *Phonology*, 36(4):627–653, doi:10.1017/S0952675719000320.

Hugh RABAGLIATI, Brock FERGUSON, and Casey LEW-WILLIAMS (2019), The profile of abstract rule learning in infancy: Meta-analytic and experimental evidence, *Developmental Science*, 22:1–18.

Fariz RAHMAN (2016), seq2seq: Sequence to sequence learning with Keras, `https://github.com/farizrahman4u/seq2seq`.

D. Victoria RAU, Hui-Huan Ann CHANG, and Elaine E. TARONE (2009), Think or sink: Chinese learners' acquisition of the English voiceless interdental fricative, *Language Learning*, 59(3):581–621.

D.E. RUMELHART and J.L. MCCLELLAND (1986), On learning the past tenses of English verbs, in J.L. MCCLELLAND and D.E. RUMELHART, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2: Psychological and Biological Models, pp. 216–271, The MIT Press.

Mark S. SEIDENBERG and Jeff L. ELMAN (1999), Do infants learn grammar with algebra or statistics?, *Science*, 284(5413):433.

Thomas R. SHULTZ and Alan C. BALE (2001), Neural network simulation of infant familiarization to artificial sentences: Rule-like behavior without explicit rules and variables, *Infancy*, 2(4):501–536.

Mary H. SKEEL (1969), Perceptual confusions among fricatives in preschool children., Technical report, The University of Wisconsin, `https://files.eric.ed.gov/fulltext/ED036789.pdf`.

Nitish SRIVASTAVA, Geoffrey HINTON, Alex KRIZHEVSKY, Ilya SUTSKEVER, and Ruslan SALAKHUTDINOV (2014), Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, 15(1):1929–1958.

Pavol ŠTEKAUER, Salvador VALERA, and Lívia KÖRTVÉLYESSY (2012), *Word-formation in the world's languages: a typological survey*, Cambridge University Press.

Joseph Paul STEMBERGER and Marshall LEWIS (1986), Reduplication in Ewe: Morphological accommodation to phonological errors, *Phonology*, 3:151–160.

Ilya SUTSKEVER, Oriol VINYALS, and Quoc V. LE (2014), Sequence to sequence learning with neural networks, in *Advances in neural information processing systems*, pp. 3104–3112.

Tijmen TIELEMAN and Geoffrey HINTON (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.

Guillermo VALLE-PEREZ, Chico Q CAMARGO, and Ard A. LOUIS (2018), Deep learning generalizes because the parameter-function map is biased towards simple functions, in Yoshua BENGIO and Yann LECUN, editors, *Proceedings of the 6th International Conference on Learning Representations*.

Rachelle WAKSLER (1999), Cross-linguistic evidence for morphological representation in the mental lexicon, *Brain and Language*, 68(1–2):68–74.

Yang WANG (2021), Recognizing reduplicated forms: Finite-state buffered machines, in Garrett NICOLAI, Kyle GORMAN, and Ryan COTTERELL, editors, *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pp. 177–187.

Janet F. WERKER and Richard C. TEES (1983), Developmental changes across childhood in the perception of non-native speech sounds, *Canadian Journal of Psychology/Revue Canadienne de Psychologie*, 37(2):278–286.

Colin WILSON (2019), Re (current) reduplication: Interpretable neural network models of morphological copying, *Proceedings of the Society for Computation in Linguistics (SCiL)*, 2:379–380.

Brandon Prickett
0000-0001-9217-2130
bprickett@umass.edu

Aaron Traylor
0000-0002-0975-1914
aaron_traylor@brown.edu

Joe Pater
0000-0002-4784-3799
pater@linguist.umass.edu

Department of Computer Science,
Brown University

Department of Linguistics,
University of Massachusetts Amherst

Brandon Prickett, Aaron Traylor, and Joe Pater (2022), *Learning reduplication with a neural network that lacks explicit variables*, Journal of Language Modelling, 10(1):1–38
https://dx.doi.org/10.15398/jlm.v10i1.274