Learning Stress Patterns with a Sequence-to-Sequence Neural Network

Brandon Prickett
University of Massachusetts Amherst
bprickett@umass.edu

Joe Pater
University of Massachusetts Amherst
pater@umass.edu

Abstract

We present the first application of modern neural networks to the well studied task of learning word stress systems. We tested our adaptation of a sequence-to-sequence network on the Tesar and Smolensky test set of 124 "languages", showing that it acquires generalizable representations of stress patterns in a very high proportion of runs. We also show that it learns restricted lexically conditioned patterns, known as stress windows. The ability of this model to acquire lexical idiosyncracies, which are very common in natural language systems, sets it apart from past, non-neural models tested on the Tesar and Smolensky data set.

1 Introduction

Some of the earliest work in computational phonology investigated the acquisition and representation of word stress patterns (Dresher and Kaye, 1990; Gupta and Touretzky, 1994). Stress is of interest because the extent of typological variation is relatively well understood, and because learning the patterns is non-trivial in various ways. A considerable amount of more recent work has focused on a data set created by Tesar and Smolensky (2000, henceforth *TS*); see further Jarosz (2013), Jarosz (2015) and Boersma and Pater (2016). The data set includes 124 languages that can be represented using 12 relatively standard Optimality Theoretic (Prince and Smolensky, 2004) constraints. Past work has tested various algorithms for weighting and ranking constraints to see which performed the best on this dataset (where performance was measured by how many of the 124 languages the models could learn with 100% accuracy).

In this paper, we explore how well a model without constraints, namely a sequence-to-sequence neural network, performs on the 124 languages. Two factors motivate this departure from constraint-based models: (i) a question of whether pre-specified structures like constraints[1] are necessary to represent and learn the stress patterns in the TS data set, and (ii) whether neural networks, which have the expressive power to capture both general and lexically specific patterns will be able to generalize stress patterns to novel data. We find that the sequence-to-sequence net does succeed in learning most of the languages, and that it generalizes to novel data, both when trained on the 124 TS languages and when trained on 6 novel patterns involving lexically conditioned stress. No previous research on the Dresher and Kaye parametric systems, or on the TS violable constraint systems, has provided a mechanism for the learning of lexically conditioned patterns – they can only acquire fully general ones. These results thus provide new challenges for future research using non-neural frameworks in this domain.

2 Background

The TS dataset was created to test an approach for handling *hidden structure* in phonology: how does a learner parse a form that it's being trained on when it hasn't learned all of the grammatical information needed for parsing in the first place? In the TS languages, this takes the form of stress patterns that are assumed to rely on foot-based structure to place the primary and secondary stress in a word. While the training data for a language includes mappings between underlying forms (strings of light and heavy syllables) to correctly stressed surface forms, that data does not include information about where the feet occur in the correct surface forms. An example of a piece of learning data in one of the TS languages is shown in (1), with *L* representing

[1] For an approach to stress learning that involves constraints that are not pre-specified, see Hayes and Wilson (2008).

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 112-118.
Held on-line February 7-9, 2022

light syllables and *L1* representing a syllable with primary stress.

(1) /L L L/ → [L **L1** L]

This datum illustrates the ambiguity present when learning stress patterns like these in a foot-based theory—this word could contain a left-aligned foot with iambic stress like [(L L1) L] or a right-aligned foot with trochaic stress like [L (L1 L)]. Each of the 124 languages consists of 62 mappings like this. The 62 input strings are all possible combinations of *L*s and *H*s for strings of 2 to 5 syllables in length, plus strings of 6 and 7 *L*s. For each input string, the candidate set of output strings consists of all possible parsings of the syllables into unary and binary feet, including ones where syllables are left unparsed. There is a minimum of one foot for each word. One of the stresses is designated as primary, and the candidate set has all possible primary stress placements.

Each output string has a corresponding vector of constraint violations, for the 12 constraints shown in (2). Each of the 124 languages in the test set can be generated by some OT ranking of these constraints. That is, some ranking can make a parsed structure optimal that is consistent with the stress pattern in the output of the learning datum, for all 62 target mappings.

(2) Constraints from the TS Data Set (constraint definitions from Jarosz, 2013)
   a. *FtBin*: Each foot must be either bimoraic or disyllabic.
   b. *Parse*: Each syllable must be footed.
   c. *Iambic*: The final syllable of a foot must be the head.
   d. *FootNon-fin*: A head syllable must not be final in its foot.
   e. *Non-fin*: The final syllable of a word must not be footed.
   f. *WSP*: Each heavy syllable must be stressed.
   g. *WordFoot-R*: Align right edge of the word with a foot.
   h. *WordFoot-L*: Align left edge of the word with a foot.
   i. *Main-R*: Align head foot with right edge of the word.
   j. *Main-L*: Align head foot with left edge of the word.
   k. *AllFeet-R*: Align each foot with right edge of the word.
   l. *AllFeet-L*: Align each foot with left edge of the word.

TS proposed that a learner uses its current grammar to parse a form and then updates its constraint rankings according to that parse. Most subsequent work (with the exception of Jarosz, 2015) has been based on this general premise (Jarosz, 2013; Boersma and Pater, 2016).

TS found that when they ran their model 10 times on each of the 124 languages in the data set, it achieved perfect accuracy on a language 60.48% of the time. Boersma and Pater (2016) found that when they used a similar parsing strategy, but with numerically weighted constraints instead of ranked ones, and with a stochastic component in the parsing process, languages were learned fully correctly 88.63% of the time. Jarosz (2013) pushed performance on this data set even further, showing that by revising the parsing strategy, success could be achieved 94.19% of the time over 10 runs of the 124 TS languages.

The state-of-the-art on the TS data for constraint-based models (95.73%) was achieved by Jarosz (2015) whose model used a pair-wise ranking grammar with a learning algorithm inspired by expectation maximization (Dempster et al., 1977). This allowed the model to avoid the problem of parsing altogether, since it was able to sample the mappings that various constraint rankings create over the course of acquisition to see which were most likely to improve its performance.

3 Our Model

While various neural network architectures have been used in phonology, such as feedforward networks (e.g., Gupta and Touretzky, 1994; Moreton, 2012), simple recurrent networks (e.g., Hare, 1990), and convolutional neural networks (e.g., Beguš, 2020), here we focus on the sequence-to-sequence architecture (Seq2Seq Sutskever et al., 2014).

This architecture was originally constructed for machine translation, but is convenient for modeling phonological mappings since it can straightforwardly map between strings of differing lengths, needed for dealing with processes like epenthesis and deletion. This is accomplished by processing the input and output strings with separate recurrent neural networks. The input is fed into the first network (called the *encoder*), which has no output layer. The recurrent connections of the encoder

then pass information about the input to the second network (called the *decoder*), which has no input, but *does* have an output layer.

A number of studies have shown that when applied to phonological patterns, Seq2Seq networks display similar learning biases to humans (Prickett, 2019, 2021) and also generalize in a human-like way on phonological and morphological tasks (Kirov and Cotterell, 2018; Prickett et al., 2018); see Corkery et al. (2019) for some caveats.

In this paper, we test the Seq2Seq architecture with both GRU (Cho et al., 2014) and LSTM (Bengio et al., 1994) layers. While both were created to help recurrent networks learn longer dependencies (specifically by addressing the problem of *vanishing gradients*; Bengio et al., 1994), past work has found that some differences exist in the biases each kind of layer has. For example, GRU layers have been shown to be biased against learning counting-based patterns that LSTMs easily acquire (Weiss et al., 2018).

In all of the simulations presented here, the network had 2 layers each in its encoder and decoder, with 20 units in each layer, and hyperbolic tangent activation functions throughout. The learning algorithm Adam (Kingma and Ba, 2015), with a batch size of 1, was used to minimize the mean squared error between the model's output and the correct output throughout learning. We leave performing a proper grid search to determine how well our results generalize to other hyperparameter settings to future work.

4 Methods and results[2]

4.1 Original Tesar and Smolensky (2000) Languages

We first tested our model to see how well it could learn the 124 original languages in the TS data set. In each input string, a timestep for the model represented a single syllable, with a [syllable weight] feature distinguishing between light ($= -1$) and heavy ($= 1$) syllables. In the output, timesteps again represented individual syllables, with the features [stress] and [primary] used to distinguish between syllables with primary stress (values of 1 and 1, respectively), secondary stress (values of 1 and -1, respectively), and no stress (values of -1 and -1, respectively).

We ran the model with a learning rate of .0005 for 500 epochs once on each of the 124 languages in the TS set. We tried versions of the Seq2Seq architecture with both GRU and LSTM layers in them and found that both layer types achieved perfect accuracy[3] in 98.39% (122/124) of the languages. This represents the highest rate of success for any model on this test set. However, it's unclear whether the model was actually encoding generalizable information, or just memorizing the 62 mappings present in each language, which would be a fairly trivial task.

Previous research has used the constraints in (2), which are all defined to hold for any string of a particular phonological type. They provide no way of encoding a situation in which two strings of a given type behave differently, as occurs in many real languages (in "exceptions", or more generally in lexically conditioned patterns). The acquired constraint-based grammars are therefore guaranteed to generalize, though at the cost of not being able to capture lexically conditioned patterns. In what follows, we test whether our learner does learn generalizable representations of the data by including multiple tokens of each type of input string. We then turn to the question of whether it can learn lexically conditioned stress patterns, including restrictions on the distribution of lexical stress.

4.2 Generalization from Tesar and Smolensky (2000) Languages

To test whether the Seq2Seq network was learning generalizable patterns or just memorizing the mappings in each language, we introduced an extra set of "lexical" features to the inputs of the TS data set. These features were implemented as a random, base-2 label for each of the tokens in training, representing the different tokens of each mapping that one would expect in an actual language. For example, the mapping from (1) would have multiple copies in training, each of which had a unique, non-zero label in their input, as illustrated in (3). These are meant to represent multiple words in a language with three light syllables and penultimate stress (like English *banana* and *cabana*).

[2]For the software used in the simulations presented here, see https://github.com/blprickett/Neural-Network-Stress.

[3]Since the network's output takes the form of a vector of real-numbered feature values, each mapping was considered correct if every feature in every timestep of its output had the correct sign (positive or negative), given that mapping's input.

Table 1: Percent of languages with perfect accuracy in training and testing.

Tokens per Type	Training	Testing
3	86.29	44.35
6	98.39	90.32

(3) Examples of Multiple Tokens of the Same Mapping Type

a. /L L L/$_{0101}$ → [L **L1** L]
b. /L L L/$_{1111}$ → [L **L1** L]
c. /L L L/$_{0001}$ → [L **L1** L]

Crucially, each token that belonged to the same mapping type had the same output in all of these simulations (that is, there was no lexical conditioning in any of the stress patterns). We created two sets of data using this system: one that had 3 tokens for each of the 62 mapping types in each of the TS languages and one that had 6 tokens for each of the types.

We ran the GRU version of the Seq2Seq model on these two data sets with a learning rate of .0005 for 200 epochs. At the end of training, we tested the model on 62 novel pieces of data, each of which represented one of the mapping types from training, but with only 0's for the lexical label features. If the network learned a generalizable pattern from training, it should correctly map all 62 of the novel testing items. The results for training and testing on both data sets are shown in Table 1.

These results suggest that, with enough tokens per type, the Seq2Seq network *does* generalize correctly from almost all (112 out of 124) of the languages in the TS data. Additionally, the accuracy on both training and testing increased with the number of tokens per type, suggesting that a number of tokens higher than 6 might allow the model to do even better on both. Natural languages of course tend to have more than six words with a given type of stress pattern, at least for shorter words.

4.3 Languages with Lexically Conditioned Stress

The final test of our model did not directly use any of the languages from the TS data set. Instead, we used the 62 input syllable strings from the TS languages and created output stressings for them using 6 novel patterns. These patterns involved *stress windows* (Kager, 2012), meaning they allowed stress to appear on any of a set of contiguous syllables at the word edge in the output, with the syllable that's stressed in a specific word being lexically specified.

Our patterns involved two basic types of window: right aligned and left aligned. Each pattern had windows of size 2, meaning the right aligned languages always had stress on their ultimate or penultimate syllables and the left aligned languages always had stress on the first or second syllables. The other feature that varied across languages was how likely stress was to occur on each of the two syllables in a window. We created three conditions for this variable: languages in which the first syllable of a window was stressed 25% of the time and second was stressed 75%, languages in which both syllables in the window were equally likely to be stressed, and languages in which the first syllable of a window was stressed in 75% of words and the second was stressed in 25% of them.

These two variables created 6 total languages to test the model on. In every language, there were 4 tokens for each mapping type, with the proportion of first syllable/second syllable stress in types' windows being the same as the language itself. This is illustrated in (4) for the language with left-aligned windows and stress on the first syllable of the window in 25% of words.

(4) Examples of Stress Window Data

a. /L L L L/$_{0101}$ → [**L1** L L L]
b. /L L L L/$_{1111}$ → [L **L1** L L]
c. /L L L L/$_{0001}$ → [L **L1** L L]
d. /L L L L/$_{1001}$ → [L **L1** L L]

We trained the model ten times on each of these 6 languages, with a learning rate of .005, until the model reached perfect accuracy on the training data. The LSTM model was able to reach this criterion for all 6 languages, while the GRU was unable to reach it for any of them in a reasonable number of epochs (we tried a variety of values for this, running the GRU model for up to 10,000 epochs with no success). At the end of training, we tested the model on novel data that had values of zero for all of the lexical label features to see how it generalized these lexically specified patterns. Table 2 shows the results on testing data for the LSTM model (no GRU results are shown since that model never succeeded in training).

With the exception of the language with left aligned windows and stress on the second syllable 25% of the time, the model seems to generalize to novel data in a way that reflects the statistics of

Table 2: Proportion of words in each language in which the window's second syllable is stressed.

Edge of the Word	Prob. in Training	Model's Results on Testing (SD)
Left	.75	0.874 (0.15)
Right	.75	0.749 (0.22)
Left	.5	0.706 (0.23)
Right	.5	0.554 (0.27)
Left	.25	0.123 (0.19)
Right	.25	0.332 (0.26)

the language it was trained on (with perhaps a bias toward stressing the second syllable more often). These results suggest that the LSTM model not only successfully learns these patterns that involve lexically conditioned stress but also can keep track of general statistical trends in the language, as has been experimentally documented for humans (see, e.g., Ernestus and Baayen, 2003).

5 Discussion

5.1 Comparison with earlier research

Our results on the TS data set with a Seq2Seq model are comparable to the best achieved with constraint based models. It is difficult to compare directly, since the 98.39% accuracy achieved in the first set of simulations, as well as on the training data in the 6 lexical item condition, could be attributable to the model simply representing each of the individual mappings, rather than learning generalizable representations. Nonetheless, the fact that it generated the correct stress pattern for 90.32% of the unseen tokens when there were 6 tokens of each type in the training data shows that it is capable of learning these patterns in a generalizable way with a high degree of accuracy.

None of the prior research on the TS data set provided a means for representing lexical idiosyncrasy, cases where two words of the same syllable shape have different stress patterns. There is a body of prior work on constraint-based approaches to lexical idiosyncratic phonology, however. Tesar (2006) presents an approach to learning exceptions in terms of contrastive specification of underlying features, Pater et al. (2012) propose an alternative that uses constraints on Underlying Representations within a MaxEnt learning framework, Moore-Cantwell and Pater (2016) explore the use of lexically specific constraints in MaxEnt, Hughto et al. (2019) study similar lexically scaled constraints, and Nazarov (2018) presents another approach to learning lexically specific constraints. All of this work has been done on very small systems, and it is not immediately clear how well the proposals will scale up to cases with even the number of constraints in the TS test set, let alone constraint sets that are large enough to deal with the complexity of less idealized individual languages, and of a fuller typology.

5.2 Future Work

A number of avenues exist for future work. The results presented in §4.2 made the TS data set more realistic by introducing multiple lexical tokens for each type of mapping. Making the TS data even more realistic is one potential future direction, for example, by representing inputs and outputs as strings of phonemes rather than just strings of light and heavy syllables.

Another question to investigate is how well this model and previous computational models of stress deal with other patterns involving exceptionality. The stress window languages introduced in 4.3 are a step in this direction, but more complex patterns of lexically conditioned stress could be explored. The constraint-based models previously tested on the TS data set had no way to represent lexical information, so equipping these simpler models with a way to handle such patterns (with, e.g., *lexically indexed constraints*; Pater, 2009) could also be fruitful.

A limitation of the TS data set is that it is based on a factorial typology of constraints rather than a real-world typology of stress-based patterns. Future work should sort through these artificially constructed languages to see which of them have real-world counterparts and which are unattested. At that point, looking closer at the learning difficulty across languages might help to explain why some are absent from the typology. Gupta and Touretzky (1994) provide an analysis of their learning results with a neural model that gives an example of how this research could proceed.

Finally, computational phonology often involves comparing predictions made by models to human behavior in artificial language learning studies (e.g., Wilson, 2006). Such studies involving stress patterns *do* exist (e.g., Carpenter, 2016), and future work should compare the aquisition and generalization observed in them to that of computational models of stress learning.

5.3 Conclusions

In this paper, we presented results showing that a Seq2Seq neural network can successfully learn a variety of stress patterns. Using the Tesar and Smolensky (2000) data set (a commonly cited benchmark for models of stress), we were able to show that the network outperformed past models when tested on how many of the languages in the data set it could acquire perfectly.

We then created an extension of this data set that included multiple tokens of each relevant mapping type in the 124 languages, and differentiated these tokens using lexically specific labels for each word. When the model was given data that included six tokens for each mapping type from the original data set, its performance on novel test items was comparable to past, state-of-the-art approaches.

Finally, we showed that the LSTM-based model could successfully learn lexically-conditioned patterns involving stress windows (Kager, 2012), something that past constraint-based models of hidden structure do not have the expressive power to do.

Taken together, these results show that (i) prespecified constraints are not necessary for a model to succesfully learn and generalize stress-based patterns and (ii) while the neural network we used had the ability to simply memorize the mappings we were training it on, it instead learned a general pattern for most languages that could be applied to novel forms.

Acknowledgements

We would like to thank the UMass Sound Workshop, as well as the audiences of the 2019 Manchester Phonology Meeting and the 2021 Annual Meeting on Phonology for helpful discussion of topics related to this paper. This research was supported by the National Science Foundation grant BCS 1650957.

References

Gašper Beguš. 2020. Modeling unsupervised phonetic and phonological learning in generative adversarial phonology. *Proceedings of the Society for Computation in Linguistics*, 3(1):138–148.

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.

Paul Boersma and Joe Pater. 2016. Convergence properties of a gradual learner in Harmonic Grammar. In John J. McCarthy and Joe Pater, editors, *Harmonic Grammar and Harmonic Serialism*, pages 389–434. Equinox Publishing, Bristol, Connecticut.

Angela C Carpenter. 2016. The role of a domain-specific language mechanism in learning natural and unnatural stress. *Open Linguistics*, 2(1).

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Maria Corkery, Yevgen Matusevych, and Sharon Goldwater. 2019. Are we there yet? Encoder-decoder neural networks as cognitive models of English past tense inflection. *arXiv:1906.01280 [cs]*. ArXiv: 1906.01280.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

B. Elan Dresher and Jonathan D. Kaye. 1990. A computational learning model for metrical phonology. *Cognition*, 34(2):137–195.

Mirjam Ernestus and R Harald Baayen. 2003. Predicting the unpredictable: Interpreting neutralized segments in dutch. *Language*, pages 5–38.

Prahlad Gupta and David S. Touretzky. 1994. Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive science*, 18(1):1–50.

Mary Hare. 1990. The role of trigger-target similarity in the vowel harmony process. In *Annual Meeting of the Berkeley Linguistics Society*, volume 16, pages 140–152.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic inquiry*, 39(3):379–440.

Coral Hughto, Andrew Lamont, Brandon Prickett, and Gaja Jarosz. 2019. Learning Exceptionality and Variation with Lexically Scaled MaxEnt. Publisher: University of Massachusetts Amherst.

Gaja Jarosz. 2013. Learning with hidden structure in optimality theory and harmonic grammar: Beyond robust interpretive parsing. *Phonology*, 30(1):27–71.

Gaja Jarosz. 2015. Expectation driven learning of phonology. *Ms., University of Massachusetts Amherst*.

René Kager. 2012. Stress in windows: Language typology and factorial typology. *Lingua*, 122(13):1454–1493.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, Conference Track Proceedings*.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Claire Moore-Cantwell and Joe Pater. 2016. Gradient Exceptionality in Maximum Entropy Grammar with Lexically Specific Constraints. *Catalan Journal of Linguistics*, 15:53.

Elliott Moreton. 2012. Inter-and intra-dimensional dependencies in implicit phonotactic learning. *Journal of Memory and Language*, 67(1):165–183.

Aleksei Nazarov. 2018. Learning within- and between-word variation in probabilistic OT grammars. *Proceedings of the Annual Meetings on Phonology*, 5.

Joe Pater. 2009. Morpheme-specific phonology: Constraint indexation and inconsistency resolution. In Steve Parker, editor, *Phonological Argumentation: Essays on Evidence and Motivation*. Equinox, London.

Joe Pater, Robert Staubs, Karen Jesney, and Brian Smith. 2012. Learning probabilities over underlying representations. In *Proceedings of the Twelfth Meeting of the ACL-SIGMORPHON: Computational Research in Phonetics, Phonology, and Morphology*, pages 62–71.

Brandon Prickett. 2019. Learning biases in opaque interactions. *Phonology*, 36(4):627–653.

Brandon Prickett. 2021. Modelling a subregular bias in phonological learning with recurrent neural networks. *Journal of Language Modelling*, 9(1).

Brandon Prickett, Aaron Traylor, and Joe Pater. 2018. Seq2seq Models with Dropout can Learn Generalizable Reduplication. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 93–100.

Alan Prince and Paul Smolensky. 2004. *Optimality Theory: Constraint interaction in generative grammar*. Blackwell.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Bruce Tesar. 2006. Faithful Contrastive Features in Learning. *Cognitive Science*, 30(5):863–903.

Bruce Tesar and Paul Smolensky. 2000. *Learnability in optimality theory*. Mit Press.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision rnns for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745.

Colin Wilson. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive science*, 30(5):945–982.