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

Brandon Prickett and Joe Pater

University of Massachusetts Amherst

SCiL 2022

February 8, 2022

Introduction

- Learning stress patterns was one of the earliest topics in computational phonology (e.g., Dresher and Kaye, 1990; Gupta and Touretzky, 1994).
  - Stress patterns are interesting because they involve *hidden structure*
  - i.e., learners must infer structure (e.g., feet) that isn't present in a form's overt surface representation
- Tesar and Smolensky (2000) developed a dataset of 124 stress patterns
  - This has been used in previous work as a benchmark for models of phonological learning
  - However, none of these patterns involve lexically conditioned patterns—a common feature in real-world languages
- Here we test a sequence-to-sequence neural network on the 124 languages in the dataset, as well as 6 novel, lexically conditioned patterns. We want to answer the following questions:
  - **Are explicit constraints, given to a model at the beginning of learning, necessary for success on the TS languages?**
  - **Given the ability to learn and represent lexically conditioned patterns, will a model generalize to novel data or just memorize mappings?**

Background

Hidden Structure and Learning

- In phonological learning, we are typically not given the full structure of the data.
- Footing is a common example of hidden structure: overt [babába] is compatible with at least two full structures, each violating different constraints (e.g., Trochee and Iamb).
- Given the overt forms of the language data, we have to infer hidden structures like feet, and typically this inference has to occur in parallel with learning the grammar.

/bababa/	Trochee	Iamb
(babá)ba	-1	0
ba(bába)	0	-1

The Learning Task

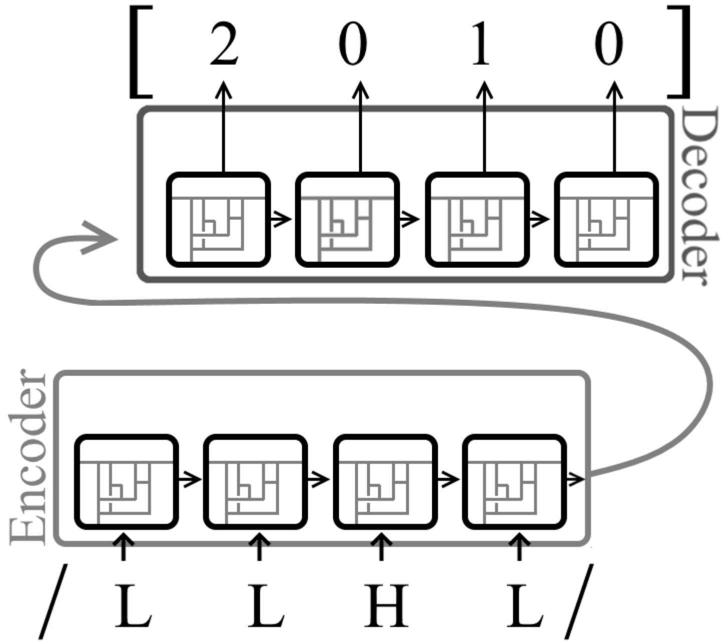
- To test how well models of phonological learning can deal with hidden structure, Tesar and Smolensky (2000; henceforth *TS*) created a dataset of 124 stress patterns.
- These patterns are a sample of the factorial typology of 12 constraints.
  - The ranking of these constraints determines various aspects of each pattern, like whether feet are iambs or trochees by default and what edge of the word the feet are aligned to.
  - Note that all 124 patterns contain no lexical conditioning—they're meant to apply uniformly to all words.
- The training data for each pattern consists of 62 underlying representations, as well as the overt surface form that the pattern maps that UR to.
  - URs are sequences of unstressed light and heavy syllables, overt SRs are the same sequence, with primary and secondary stress marked on the word.
  - Tesar and Smolensky also provide constraint violations for all possible foot placements in each surface form, however we ignore those here since our neural network doesn't explicitly use constraints or feet.
- Past work has judged models by what proportion of the 124 languages they're able to get 100% accuracy on (i.e., all 62 URs map to the correct over surface form).

Past Models

- TS introduced *Robust Interpretive Parsing* (*RIP*) to deal with hidden structure and Boersma and Pater (2008) tested RIP with a number of different constraint-based approaches on the TS dataset.
  - Out of the models they tested, the most successful was a noisy, exponential version of Harmonic Grammar (Legendre et al. 1990) that was trained using the *Gradual Learning Algorithm* (*GLA*; Boersma 1997, Jesney 2007).
  - On average, the model succeeded on **89.95%** of the patterns (after being run on all 124 of them 10 times).
- Jarosz (2013) improved upon RIP with *Expected Interpretive Parsing* (*EIP*) and applied it to a number of constraint-based models as well.
  - Her best result used a noisy Harmonic Grammar model with this new parsing strategy and was successful on **94.19%** of the languages (again, after 10 repetitions per language).
- Jarosz (2015) later showed that even better performance could be achieved with *Probabilistic Ranking Grammars*, optimized using *Batch Expectation Driven Learning* (Jarosz 2015).
  - This model was successful on the TS languages an average of **95.73%** of the time, over 10 reps.

Our Model

- Sequence-to-sequence neural networks were originally designed for machine translation (Sutskever et al., 2014), but are convenient for modelling phonological mappings.
  - They're made up of two connected recurrent neural networks: the *encoder* which processes the input and the *decoder* which produces the output.
  - Sequence-to-Sequence networks have been shown to have human-like learning/generalization (Prickett 2019; Kirov and Cotterell 2018; Prickett 2021) when trained on phonological and morphological tasks.
- The network represents the input sequence as a series of timesteps, each of which is a vector of features with real-numbered values.
  - For the simulations here, input syllables had one feature, showing whether they were light (-1) or heavy (1).
  - Output syllables had two features: one showed whether the syllable was stressed (1) or not (-1) and one showed whether that stress was primary (1) or secondary (-1).



Methods and Results

The Original TS Dataset

- As a first test of our model, we ran it on the original 124 languages in the TS dataset.
  - Training data for each language were the relevant 62 mappings from a sequence of light/heavy syllables to a sequence of syllables with primary/secondary/no stress.
  - For example: **/L L L/ → [L L1 L]**
  - No testing data was present for these runs, since the original TS dataset isn't meant to measure a model's ability to generalize to novel data.
- Hyperparameters:
  - Learning rate: .0005
  - Learning duration: 500 epochs (i.e., 500 full passes through the training data)
  - Batch size: 1 (i.e., online learning)
- For a language to be considered successful, the model had to map all 62 inputs from training to outputs in which each feature has the correct sign (positive/negative).
  - With both GRU and LSTM hidden layers, we found that the model succeeded on **98.39% (122/124)** of the TS patterns.

Limitations of the TS Data

- The 62 data points in each of the TS patterns are meant to represent word types, rather than tokens.
  - Each type of word is meant to have the same kind and number of syllables underlyingly and map to the same stress pattern on the surface (e.g., *banana* and *cabana* in English are both /L L L/→ [0 1 0] mappings).
- This is sufficient for testing all of the earlier constraint-based models, since the constraints are completely general—that is, they apply to any phonological sequence that violates them.
- However, neural networks have the ability to learn both general processes and exceptions to those processes.
  - This allows our model to learn lexically conditioned patterns—a typologically common phenomenon that we'll return to shortly.
  - But it also means that it could have just memorized the 62 mappings in each pattern rather than learning anything generalizable.

Generalizing from the TS Patterns

- To overcome this limitation, we randomly produced versions of each of the 124 languages that have multiple tokens per type.
  - That is, for each of the 62 kinds of mappings, our new language had multiple forms that mapped in the same way.
- We differentiated the different tokens by adding a set of features to each timestep that represented a unique lexical ID for that token.
- We trained the model with GRU layers in the same way as before and then tested it on data with lexical ID's of 0:

$$/\mathrm{L\ L\ L}/_{0101} \rightarrow [\mathrm{L}\ \mathbf{L1}\ \mathrm{L}]$$
$$/\mathrm{L\ L\ L}/_{1111} \rightarrow [\mathrm{L}\ \mathbf{L1}\ \mathrm{L}]$$
$$/\mathrm{L\ L\ L}/_{0001} \rightarrow [\mathrm{L}\ \mathbf{L1}\ \mathrm{L}]$$

Tokens per Type	Training	Testing
3	86.29%	44.35%
6	98.39%	**90.32%**

Languages with Lexically Conditioned Stress

- So far, we’ve seen that the sequence-to-sequence neural network can succeed on the TS dataset—even if you only count languages that it was able to generalize from, it’s comparable to many past models that were tested on these patterns.
- However, our model is also able to do something that none of those models had the expressive power to accomplish: lexically conditioned patterns.
  - That is, patterns in which individual datapoints are exceptions to a more general rule, or when arbitrary classes of words follow different rules.
- While methods for representing exceptionality in constraint-based frameworks exist (e.g., Pater 2009; Nazarov 2018; Hughto et al. 2019), to our knowledge, none of these have been used to learn the TS languages.

TS-Style Stress Window Patterns

- To demonstrate our model's ability to learn lexically conditioned patterns and to see how it generalizes from such patterns, we adapted the 62 input forms from the TS dataset to create six novel languages.
- These languages all made use of *stress windows,* meaning they allowed stress to appear on any pair of contiguous syllables at the word edge, with the syllable that's stressed in a specific word being lexically specified.
- We varied the languages along two dimensions:
  - Which word edge the window appeared on (left or right)
  - With what probability stress appeared on the first syllable in the window (.25, .5, .75)
- We again used lexical ID features to uniquely mark each form, however these were now crucial for the model to know whether a form stressed the first or second syllable in its window.

$$/\mathrm{L\ L\ L\ L}/_{0101} \rightarrow [\mathbf{L1}\ \mathrm{L\ L\ L}]$$

$$/\mathrm{L\ L\ L\ L}/_{1111} \rightarrow [\mathrm{L}\ \mathbf{L1}\ \mathrm{L\ L}]$$

$$/\mathrm{L\ L\ L\ L}/_{0001} \rightarrow [\mathrm{L}\ \mathbf{L1}\ \mathrm{L\ L}]$$

$$/\mathrm{L\ L\ L\ L}/_{1001} \rightarrow [\mathrm{L}\ \mathbf{L1}\ \mathrm{L\ L}]$$

Stress Window Results

- We ran the network on these stress window patterns for as many epochs as it took for it to achieve 100% accuracy on the training data.

- When using GRU layers in the model, we never saw it reliably converge on these languages, despite running it for 10,000 epochs on each one.
  - We're unsure as to why GRU layers struggled with stress windows, although it might be because of their bias against counting-based patterns (Weiss et al. 2018).

- However, with LSTM layers, the model converged reliably for every language. At the end of training, we gave it data with lexical ID's of 0. Results below show pr(stressing 1st syllable in the window) for novel forms:

In Training	Left Edge of the Word	Right Edge of the Word
.25	0.126 (.15)	0.251 (.22)
.5	0.294 (.22)	0.446 (.27)
.75	0.877 (.19)	0.668 (.26)

Discussion

Future Work

- Making the TS dataset more realistic. The lexical IDs were a step in this direction, however one could take this in many different directions:
  - Using phonemes as timesteps (rather than syllables)
  - Having noise in training (meant to represent speech errors)
  - Distribute word lengths more realistically (instead of allowing long words to be just as common as shorter ones)
- Comparing this model with a constraint-based approach that can express lexical patterns.
  - Would those constraint-based approaches be able to generalize to novel data as reliably as the network?
  - Would they behave similarly when generalizing from stress windows?
- Exploring the TS patterns more carefully to see which are and aren’t typologically attested.
- Comparing the network’s acquisition of stress patterns to human behavior in an artificial language learning study (e.g., Carpenter 2016).

Conclusions

- **Are explicit constraints, given to a model at the beginning of learning, necessary for success on the TS languages?**
  - No, we showed that a neural network without explicit constraints could succeed on **98.39%** of the languages in the dataset, better than any past approach.
  - Even when forced to generalize to novel data, the model succeeded on **90.32%** of the patterns, comparable to past approaches.
- **Given the ability to learn and represent lexically conditioned patterns, will a model generalize to novel data or just memorize mappings?**
  - We demonstrated that the neural network *could* learn and represent lexically conditioned stress window patterns.
  - It also generalized the TS stress patterns to novel data (as mentioned above)...
  - ...And generalized the window-based patterns in a way that was close to frequency matching (a behavior observed in humans; see, e.g., Ernestus and Baayen 2003).

References

References Gasper Begu ˇ s. 2020. Modeling unsupervised phonetic ˇ 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 domainspecific language mechanism in learning natural and unnatural stress. Open Linguistics, 2(1).

Kyunghyun Cho, Bart Van Merrienboer, Caglar Gul- ¨ cehre, 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.

Rene Kager. 2012. Stress in windows: Language typol- ´ ogy 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 betweenword 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.

Thank you!

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.

Appendix: Tesar and Smolensky (2000) Constraints

- *WSP*: stress heavy syllables.
- *FootNonFinal*: head syllables must not come foot final.
- *Iambic*: head syllables must come foot final.
- *Parse*: Each syllables must be footed.
- *FtBin*: feet must be one heavy syllable or two syllables of either weight.
- *WordFootLeft*: align feet with the left edge of the word.
- *WordFootRight*: align feet with the right edge of the word.
- *MainLeft*: align the head foot with the left edge of the word.
- *MainRight*: align the head foot with the right edge of the word.
- *AllFeetLeft*: align all feet with the left edge of the word.

  AllFeetRight: align all feet with the right edge of the word.
- *NonFinal*: the final syllable in a word must not be footed.

Appendix: Epochs to Convergence for Stress Window Patterns

Condition: side-pr(stressing 1st syll)	Mean	SD
Left-0.25	211	94.3
Left-0.5	281	124.9
Left-0.75	191	53.4
Right-0.25	211	53.4
Right-0.5	211	70.0
Right-0.75	251	67.1