# RNNs and Subregular Formal Languages

**The Linguists**
Cody St. Clair, Joanne Chau and Emily Peterson

Stony Brook University

# Problem Statement

**What are neural networks actually learning?**

# Problem Statement

**What are neural networks actually learning...**

**in terms of Subregular Formal Languages?**

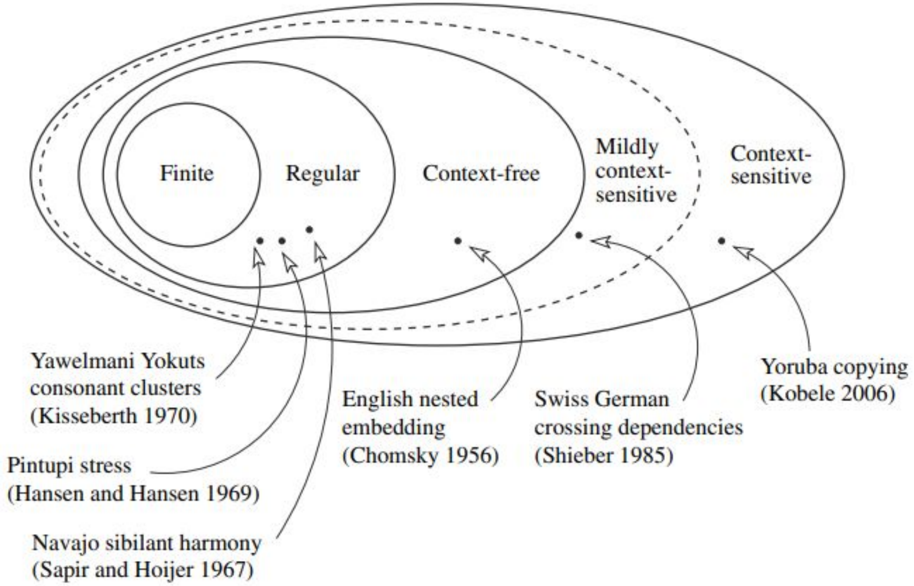



**Figure 1**
The Chomsky hierarchy

(Heinz 2010)

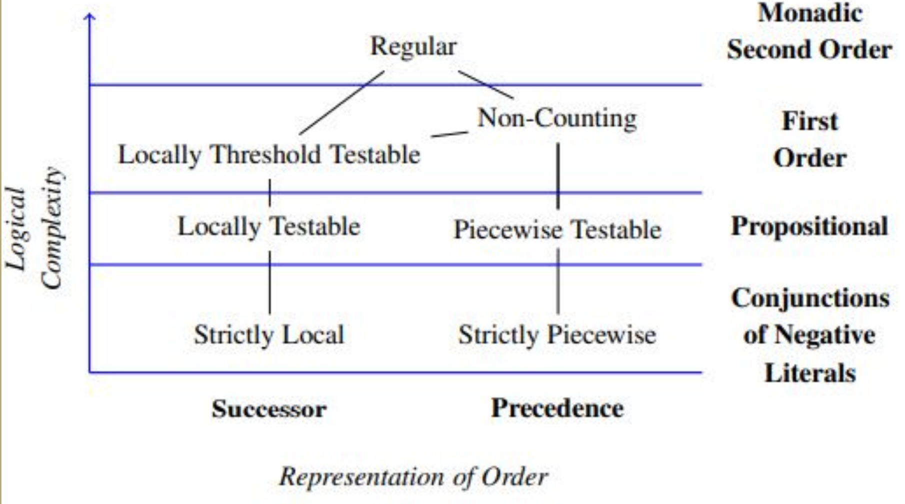



Figure 1: Subregular language classes with inclusion shown from the top down.

(Avcu et al 2010)

Stony Brook University

# How has this problem been addressed?

Previous work has investigated the performance of simple RNNs and LSTMs on **long-distance dependencies**

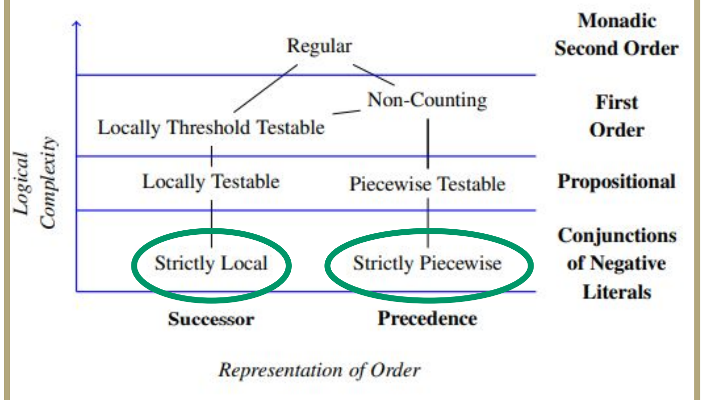



Figure 1: Subregular language classes with inclusion shown from the top down.

**Subregular Complexity and Deep Learning**


**Enes Avcu**
Department of Linguistics and Cognitive Science
University of Delaware
enesavc@udel.edu

**Chihiro Shibata**
School of Computer Science
Tokyo University of Technology
shibatachh@stf.teu.ac.jp

**Jeffrey Heinz**
Department of Linguistics
Institute of Advanced Computational Science
Stony Brook University
jeffrey.heinz@stonybrook.edu



**Abstract**

This paper argues that the judicial use of formal language theory and grammatical inference are invaluable tools in understanding how deep neural networks can and cannot represent and learn long-term dependencies in temporal sequences.

Learning experiments were conducted with two types of Recurrent Neural Networks (RNNs) on six formal languages drawn from the Strictly Local (SL) and


learning networks (Goodfellow et al., 2016) are able to learn. The main ideas are illustrated with experiments testing how well two types of Recurrent Neural Networks (RNNs) can learn different kinds of simple, subregular formal languages with a grammatical inference algorithm serving as a baseline.

Using formal languages to investigate the learning capabilities of neural networks is not without precedent. Much earlier research also used formal languages to probe the learning capabilities of neural networks; Schmidhuber (2015, sec. 5.13)

(Avcu et al 2010)

# Our contribution

A scalable **workflow** for investigating the performance of **more models** on **more subregular classes**

| Model | Classes |
| --- | --- |
| → **GRU** | → **SL, SP** |
| → **BiGRU** | → **Tier-Based Strictly Local** |
| → LSTM | |
| → BiLSTM | → LT, LTT, SF, R, |
| * Dropout | … |

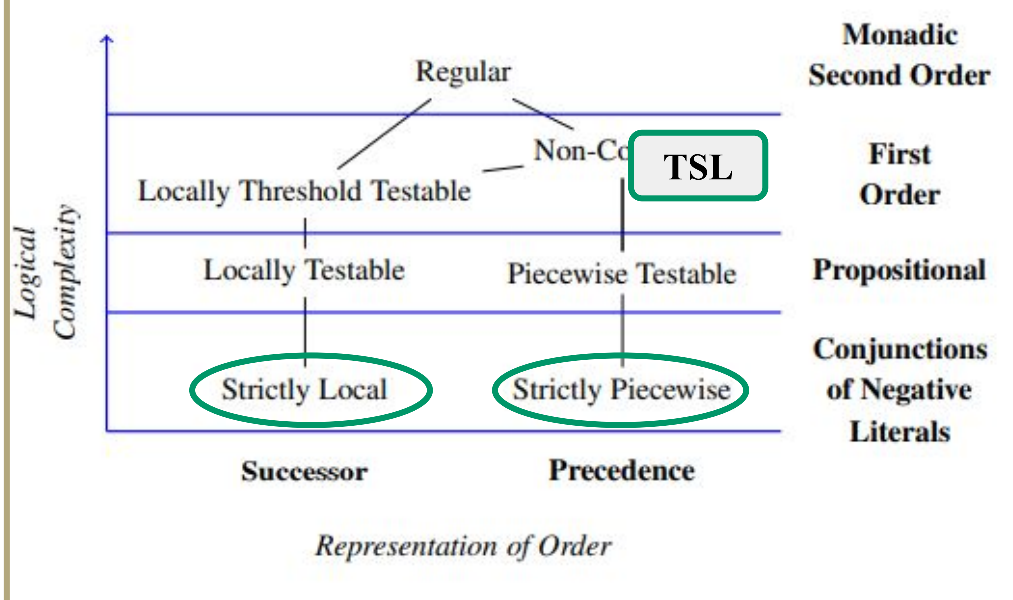


Figure 1: Subregular language classes with inclusion shown from the top down.

(Avcu et al 2010)

# The specific issue: how do NNs do on TSL?

- **SL:** intervocalic voicing; if there is a consonant between two vowels, that consonant *must* be voiced

aba → Accepted

apa → Declined

- **SP:** Samala Sibilant Harmony; no word may contain the sounds [s] and [ʃ] regardless of how far apart they are

haʃxintilawaʃ → Accepted

sishuleqpeyus → Declined

hasxintilawaʃ → Declined

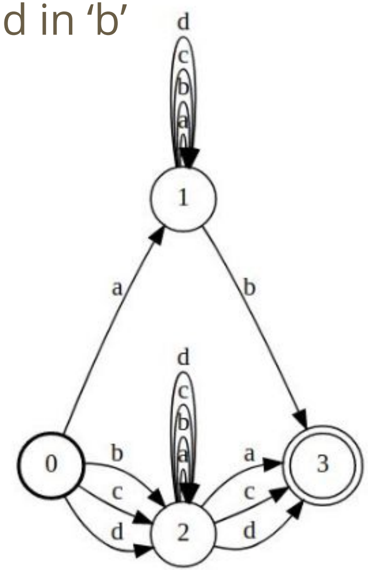
sishuleqpeyuʃ → Declined

# The specific issue: how do NNs do on TSL?

- **TSL:**

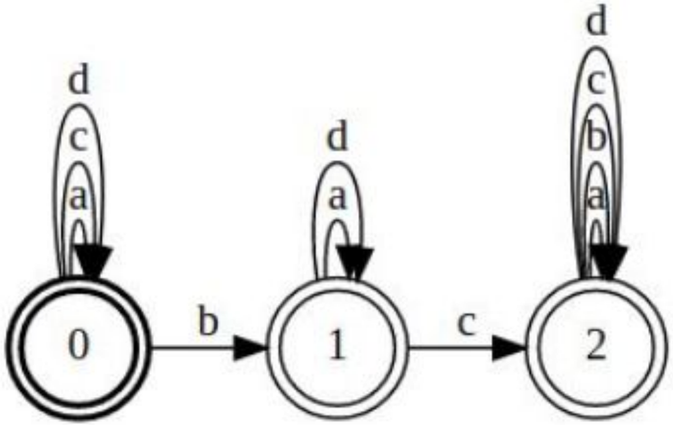
Circumfixation;

Strings starting in ‘a’ must end in ‘b’




Long-distance dissimilation;

‘b’ cannot appear after a ‘b’ unless a ‘c’ intervenes




(Aksënova et al 2016)

# Challenges: Data Generation

180 datasets;
36 models;

| 2 | x | 3 | x | 3 | x | 2 | x | 3 | = 108 |
|---|---|---|---|---|---|---|---|---|---|
| architectures | | language classes | | languages per class | | set sizes | | tests | **tests** |

language classes:

SL
SP
TSL

set sizes:

10k
100k

tests:

1) short strings
2) long strings
3) adversarial pairs

# What is the specific system you are using?

- **Data generation**
  - OpenFST
  - Pynini
- **Models**
  - Binary classification (in the language vs not)
  - Subclassed Tensorflow; GRU, BiGRU, (LSTM, BiLSTM)
  - Dropout
  - Character embeddings, size 100
  - 30 epochs
- **Evaluation**
  - TensorBoard
  - Accuracy & F-scores

# Initial Results

## GRU, no dropout

| | | | Unidirectional | | | | | | Bidirectional | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SL.4.2.1 | | SL.4.2.2 | | SL.4.2.4 | | SL.4.2.1 | | SL.4.2.2 | | SL.4.2.4 | |
| | | | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy |
| SL | 100k | Test1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 10k | Test1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test2 | 1 | 1 | 1 | 1 | 0.9999 | 0.9999 | 1 | 1 | 1 | 1 | 0.9999 | 0.9999 |
| | | Test3 | 0.9995 | 0.9995 | 1 | 1 | 0.95111 | 0.9486 | 0.9995 | 0.9995 | 1 | 1 | 0.95111 | 0.9486 |
| | | | SP.4.2.1 | | SP.4.2.2 | | SP.4.2.4 | | SP.4.2.1 | | SP.4.2.2 | | SP.4.2.4 | |
| | | | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy |
| SP | 100k | Test1 | 1 | 1 | 1 | 1 | 1 | 0.99999 | 1 | 1 | 1 | 1 | 1 | 0.99999 |
| | | Test2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test3 | 1 | 1 | 1 | 1 | 0.80633 | 0.75982 | 1 | 1 | 1 | 1 | 0.80633 | 0.75982 |
| | 10k | Test1 | 1 | 1 | 0.9999 | 0.9999 | 0.9989 | 0.9989 | 1 | 1 | 0.9999 | 0.9999 | 0.9989 | 0.9989 |
| | | Test2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test3 | 1 | 1 | 0.67074 | 0.5091 | 0.76628 | 0.695 | 1 | 1 | 0.67074 | 0.5091 | 0.76628 | 0.695 |
| | | | TSL.0 | | TSL.1 | | TSL.2 | | TSL.0 | | TSL.1 | | TSL.2 | |
| | | | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy |
| TSL | 100k | Test1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test2 | 1 | 1 | 1 | 1 | 1 | 0.99997 | 1 | 1 | 1 | 1 | 1 | 0.99997 |
| | | Test3 | 1 | 1 | 1 | 1 | 0.98674 | 0.98656 | 1 | 1 | 1 | 1 | 0.98674 | 0.98656 |
| | 10k | Test1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test2 | 1 | 1 | 1 | 1 | 0.9999 | 0.9999 | 1 | 1 | 1 | 1 | 0.9999 | 0.9999 |
| | | Test3 | 1 | 1 | 1 | 1 | 0.98678 | 0.9866 | 1 | 1 | 1 | 1 | 0.98678 | 0.9866 |

# Initial Results

GRU, no dropout

| Unidirectional | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | SL.4.2.1 | | SL.4.2.2 | | SL.4.2.4 | |
| | | | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy |
| SL | 100k | Test1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test2 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test3 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 10k | Test1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test2 | 1 | 1 | 1 | 1 | 0.9999 | 0.9999 |
| | | Test3 | 0.9995 | 0.9995 | 1 | 1 | 0.95111 | 0.9486 |
| | | | SP.4.2.1 | | SP.4.2.2 | | SP.4.2.4 | |
| | | | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy |
| SP | 100k | Test1 | 1 | 1 | 1 | 1 | 1 | 0.99999 |
| | | Test2 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test3 | 1 | 1 | 1 | 1 | 0.80633 | 0.75982 |
| | 10k | Test1 | 1 | 1 | 0.9999 | 0.9999 | 0.9989 | 0.9989 |
| | | Test2 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test3 | 1 | 1 | 0.67074 | 0.5091 | 0.76628 | 0.695 |
| | | | TSL.0 | | TSL.1 | | TSL.2 | |
| | | | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy |
| TSL | 100k | Test1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test2 | 1 | 1 | 1 | 1 | 1 | 0.99997 |
| | | Test3 | 1 | 1 | 1 | 1 | 0.98674 | 0.98656 |
| | 10k | Test1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | Test2 | 1 | 1 | 1 | 1 | 0.9999 | 0.9999 |
| | | Test3 | 1 | 1 | 1 | 1 | 0.98678 | 0.9866 |

- Bidirectionality had no effect on test predictions and scores
- The GRUs learned the patterns perfectly in most cases
- The GRUs performed the worst on the more complicated SP languages

# Initial Conclusions

- The TSL languages did not challenge the GRUs more than the SL languages
- Complex SP languages are indeed a challenge to GRUs
- “We find that the model’s ability to generalize this structure beyond the training distribution depends greatly on the chosen random seed, even when performance on the standard test set remains the same.”
  Weber et al (2018)

To Be Continued

# References

Aksënova, A., Graf, T., & Moradi, S. (2016, August). Morphotactics as tier-based strictly local dependencies. In Proceedings of the 14th sigmorphon workshop on computational research in phonetics, phonology, and morphology (pp. 121-130).

Avcu, E., Shibata, C., & Heinz, J. (2017). Subregular complexity and deep learning. arXiv preprint arXiv:1705.05940.

Gorman, K. (2016, August). Pynini: A Python library for weighted finite-state grammar compilation. In Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata (pp. 75-80).

Heinz, J. (2010). Learning long-distance phonotactics. Linguistic Inquiry, 41(4), 623-661.

Heinz, J., Rawal, C., & Tanner, H. G. (2011, June). Tier-based strictly local constraints for phonology. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies (pp. 58-64).

Weber, N., Shekhar, L., & Balasubramanian, N. (2018). The fine line between linguistic generalization and failure in Seq2Seq-attention models. arXiv preprint arXiv:1805.01445.